

SAND REPORT

SAND2002-1015
Unlimited Release
Printed April 2002

Evaluation Techniques and Properties of an Exact Solution to a Subsonic Free Surface Jet Flow

Allen C. Robinson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/ordering.htm>



SAND2002-1015
Unlimited Release
Printed April 2002

Evaluation Techniques and Properties of an Exact Solution to a Subsonic Free Surface Jet Flow

Allen C. Robinson
Computational Physics Research and Development
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-0819

Abstract

Computational techniques for the evaluation of steady plane subsonic flows represented by Chaplygin series in the hodograph plane are presented. These techniques are utilized to examine the properties of the free surface wall jet solution. This solution is a prototype for the shaped charge jet, a problem which is particularly difficult to compute properly using general purpose finite element or finite difference continuum mechanics codes. The shaped charge jet is a classic validation problem for models involving high explosives and material strength. Therefore, the problem studied in this report represents a useful verification problem associated with shaped charge jet modeling.

Acknowledgments

J. D. Cole pointed out the work of Nieuwland. D. E. Amos, M. E. Kipp and J. M. McGlaun of Sandia National Laboratories provided valued expertise and insight during this work. Chris Garasi and Peter Stoltz provided invaluable instruction with respect to the IDL graphics language. Tim Trucano and Dan Carroll assisted by reviewing a draft of this document.

Nomenclature

ρ	density
p	pressure
ν	specific volume
c	sound speed
e	specific internal energy
γ	power coefficient in isentropic equation of state relation
Γ	Grüneison coefficient
q	velocity magnitude or flow speed
q_∞, ρ_∞	subscript ∞ denotes a free streamline value in dimensional units
c_0, ρ_0	subscript 0 denotes stagnation point value in dimensional units
q_1, ρ_1	subscript 1 denotes a free streamline value in non-dimensional units
q_{max}	maximum speed
q_{cr}	critical speed
τ	$(q/q_{max})^2$
θ	angular flow direction
M	Mach number
u	$q \cos \theta$
v	$q \sin \theta$
ϕ	velocity potential
ψ	stream function
$\psi_n(\tau)$	fundamental Chaplygin function
$F_n(\tau)$	Chaplygin function given by Gauss hypergeometric series
β	incoming jet angle
a_n, b_n	useful parameters related F_n
W	complex potential
Ω	complex hodograph variable = $(q/q_1)e^{-i\theta}$
Θ	generic phase offset in series representations

1 Introduction

Transient dynamic continuum mechanics codes can be used to analyze the effects of explosive-metal interaction and ballistic penetration events. These general purpose codes allow the use of many materials and complex configurations. The complexity of such codes is such that it is extremely important to test the results, methodologies and applicability regions of the codes relative to exact solutions (verification) and experimental data (validation) [27]. A methodology which has been found to be very successful in some physical regime or for certain problems may fail when applied to a new class of problems. This report is concerned with the detailed description of a steady plane isentropic subsonic jet impinging on a flat wall. The problem is a prototype for the formation of a shaped charge jet and is a high strain and strain rate flow. This work was briefly summarized in Chapter 16 of Volume 2 of Avner Friedman's series *Mathematics in Industrial Problems* after the present author's presentation to the Institute for Mathematics and Its Applications on May 19, 1989 [12]. Friedman proposed several mathematical problems in cylindrical coordinates in the same chapter. Due to current interest in the computational science and engineering community in issues of verification and validation of computational simulations, it seems useful to make a full accounting of this work.

A conical shaped charge consists of a cylinder of high explosive containing a hollowed out cone surfaced with a metal liner. The detonation products collapse the liner and a high velocity metallic jet is formed. During this process the jet heats due to shock loading and plastic work [36, 26, 32]. It is widely believed that typical copper lined shaped charges form jets of material in the solid state. Historically, this was substantiated mainly by observation of the solid fracture characteristics observed in jet breakup. Confirmation of the existence of a solid state for aluminum jets and on the surface of copper jets has been made from x-ray diffraction patterns [16, 17]. For copper jets temperature measurements in the $400 - 600^\circ C$ range, which are well below the melt temperature of $1080^\circ C$, have been made based on two-color IR radiometry [15]. Jet particles in the solid state have also been recovered using soft catch techniques [35].

The shaped charge jet problem during the quasi-steady collapse phase may be idealized with a steady compressible fluid model. Key features in this model include large velocity gradients in small spatial regions as well as very large strains in a steady subsonic isentropic free-surface flow. The features

combine to generate computational difficulties with the shaped charge jet problem. The shaped charge jet is very difficult to model correctly by either a Lagrangian finite element code or an Eulerian code. Lagrangian codes tend to experience severe deformation in the jet leading to a breakdown of the numerical method due to element distortion. The Eulerian codes may have difficulty with interfaces and excessive heating of jet material. This problem also represents a reasonably severe test of arbitrary Lagrangian-Eulerian (ALE) and h and h-p adaptive modeling capabilities.

The above mentioned heating problem for Eulerian codes may include unrealistic temperature diffusion into the liner from the explosive products, unphysical numerical exchange of kinetic energy to internal energy [19, 28] and heating due to artificial viscosity terms in high compression rate shockless processes [24]. Variations in numerical algorithms can produce dramatic differences in estimates of internal energy and temperature. The question is really one of entropy. In any numerical calculation one wishes any excess numerical production of entropy to be much smaller than the correct entropy increase. The numerical difficulties may be particularly acute when the flow to be computed is isentropic. If confidence is to be placed in calculations which purport to include advanced material modeling, it is necessary to develop reliable numerical methods and practical calculational rules of thumb to deal with the shaped charge jetting problem in the case of simple hydrodynamic material modeling. For example, temperature dependent yield and fracture models require that heating in flows with or without shocks as well as heating due to plastic work be calculated accurately. Mechanical response is affected by solid-solid, solid-liquid and liquid-vapor phase transitions and these transitions will appear in the numerical simulation correctly only if the thermodynamic state space is traversed correctly. The proper application of advanced material modeling in shaped charge simulations thus depends upon proper energy partitioning in the numerical method. In particular it may be difficult for a numerical method to distinguish a rapid shockless transition from a true shock which is to be captured by the numerical method. Of course, it does not follow that an algorithm which can effectively compute a shockless flow properly will necessarily capture shocks well. The complete shaped charge jet problem requires consistent and effective modeling for both shocks and subsonic quasi-steady state flow. This report is concerned with a specific test problem which may be used, for example, to test the capability of a shock capturing code to model shockless high-strain-rate isentropic subsonic flow.

The conical shaped charge jet has been reasonably modeled in a gross engineering sense for years by the assumption that the jet collapse process is approximately a steady state in the frame of reference of the collapse point and that free-surface jet theory can be applied [3]. Operational shaped charges collapse the liner at a subsonic velocity in order to form coherent jets. Supersonic collapse speeds result either in no jet formation or incoherent jets [8]. Steady compressible subsonic plane and axis-symmetric free surface jet flows may be effectively calculated with specialized finite difference codes employing boundary fitting coordinate systems or by computing in the hodograph plane [25, 9]. The hodograph plane uses velocity and flow angle, (q, θ) , as independent variables. However, as discussed above these same flows can still represent a significant challenge for *general purpose* transient dynamics codes. Exact solutions can be used as test cases. Karpp developed a test problem, the symmetrical impact of two plane jets, for the purpose of comparison with hydrodynamic code solutions and in order to better understand compressible jet flow [18]. He used the Chaplygin pressure-density relation given by

$$p = (\rho_\infty c_\infty)^2 (1/\rho_\infty - 1/\rho) = (\rho_\infty c_\infty)^2 (\nu_\infty - \nu) \quad (1)$$

where p is pressure, ρ_∞ is the reference or free surface density, $\nu = 1/\rho$ is the specific volume, and c_∞ is the reference sound speed. A material with the above response is often termed a Chaplygin gas. The Chaplygin gas has the well-known property that the hodograph plane equations of motion can be manipulated to give the incompressible equations of motion for which standard incompressible methods apply. Thus any free-surface flow which can be solved by the usual methods of incompressible plane flow analysis can be solved for the Chaplygin gas. Karpp's work was used to assist in verifying a version of the HELP code which conserved internal energy instead of total energy in the remap step of the calculation [19].

The two parameters of the Chaplygin gas can be chosen to match any reference sound speed and pressure to give a linear curve in $p - \nu$ space. It is desirable to have an additional test problem for which the pressure-volume relation is concave upward. This is not simply an academic extension since curvature in the $p - \nu$ relation is necessary for heat addition in a shock process. Extremely high strain rate isentropic processes may have every appearance of a shock process to a finite resolution numerical grid. In order to fully test numerical methods, it appears that one is required to test with a pressure-volume relation which stiffens under compression. To this end one may chose

the isentropic relation

$$p = \bar{p}(\rho) = \kappa_\infty((\rho/\rho_\infty)^\gamma - 1) \quad (2)$$

where $\kappa_\infty \equiv \rho_\infty c_\infty^2/\gamma$ and $p(\rho_\infty) = 0$. This relation is known as the Tait or Murnaghan equation of state and is clearly of the same form as that for an ideal gas with the pressure at reference density set to zero by subtracting a constant. The Chaplygin gas is a particular case of the above relation and is chosen by setting $\gamma = -1$.

One can choose κ_∞ and γ in Equation 2 to match the first and second derivatives with respect to ρ at ρ_∞ for any given isentrope. Of course the Hugoniot may also be used, if this is more convenient, since the Hugoniot and isentrope are the same to third order in the strain. Appendix A gives a derivation for a Hugoniot which is linear in the shock velocity - particle velocity plane. It is convenient to develop a simple general equation of state relationship which matches the Murnaghan gas isentropic relations. The most obvious candidate for such an equation of state for test purposes would be a Mie-Grüneison relation for the pressure $p(e, \rho)$. In this case,

$$p(e, \rho) = \bar{p}(\rho) + \rho\Gamma(e - \bar{e}(\rho)) \quad (3)$$

where \bar{e} satisfies the isentropic differential equation for the internal energy,

$$de = -pd\nu, \quad (4)$$

so that

$$\bar{e} - e_\infty = \frac{\kappa_\infty}{\gamma - 1} \left(\frac{1}{\rho}(\rho/\rho_\infty)^\gamma - \frac{1}{\rho_\infty} \right) + \kappa_\infty(1/\rho - 1/\rho_\infty). \quad (5)$$

The Grüneison coefficient $\Gamma = \nu(\partial p/\partial e)_\nu$ is an arbitrary function of volume. For convenience, $\rho\Gamma = \alpha$ is taken to be constant. The heat capacity at constant volume, $c_\nu = (\partial e/\partial T)_\nu$, is also assumed constant. One can then derive equations for the energy, temperature, entropy and other fundamental quantities as outlined in Appendix A.

Figure 1 shows the pressure volume isentrope for a Chaplygin gas isentrope and for a Murnaghan isentrope which is matched to a standard Hugoniot relation for copper. The Mie-Grüneisen formulation using the Murnaghan isentropic relation as a reference curve represents a reasonable copper

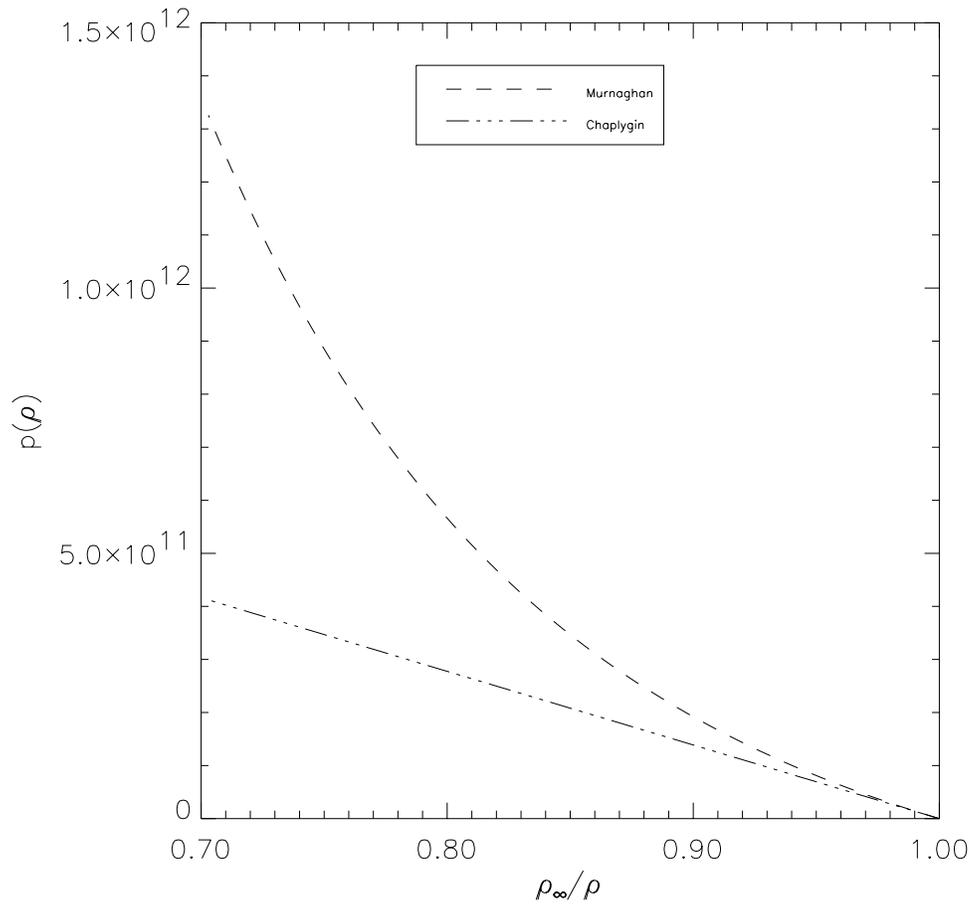


Figure 1: Comparison of Chaplygin and Murnaghan gas isentropes for Cu.

equation of state for conditions of interest and can be easily implemented as a simple equation of state model in any compressible fluid modeling code.

In this report computational procedures for evaluation of steady isentropic subsonic jet flows for the pressure-density relation of Equation 2 will be outlined. It will be seen that the steady plane irrotational compressible fluid equations of motion in the hodograph plane variables, (q, θ) , are separable and particular solutions can be obtained in terms of products of trigonometric functions and Gauss hypergeometric functions. These can be used to solve certain problems of a particular form that arise frequently in free surface flow theory. The original ideas and procedures are due to Chaplygin who solved the problem of a plane jet emerging from a slot in a wall [6]. A great many problems can be solved by Chaplygin's technique or variants of it [31]. These techniques will be applied to the solution of a plane free surface jet of subsonic velocity impinging at an angle β onto a rigid wall. Solutions are described in great detail. The basic methods outlined here will carry over in a fairly straightforward way to the evaluation of solutions of other flows of interest.

2 Steady Plane Gas Dynamics in Hodograph Variables

The theory of steady plane irrotational adiabatic compressible inviscid flow theory in the hodograph variables, (q, θ) , is well documented [2, 4, 13, 14, 22, 29]. A short summary of pertinent equations for our purposes follows below in the the notation of Bers [2]. In steady irrotational isentropic flow, with an assumed $p = p(\rho)$ relation, Bernoulli's theorem says that

$$\frac{q^2}{2} + \int \frac{dp}{\rho} = \frac{q^2}{2} + \int \frac{c^2 d\rho}{\rho} \quad (6)$$

is constant and thus gives a relation between density and flow speed. The density, sound speed, c , ($c^2 = dp/d\rho = -\rho q/\rho'(q)$), and Mach number, M , ($M^2 = -q\rho'(q)/\rho$), are then computable as a function of speed alone. For the case of Equation 2 these relationships may be given explicitly. The Bernoulli equation becomes

$$\frac{q^2}{2} + \frac{c^2}{\gamma - 1} = \frac{c_0^2}{\gamma - 1} \quad (7)$$

where the subscript zero denotes stagnation point conditions ($q = 0$). The stagnation point density and sound speed are given by

$$c_0^2 = c_\infty^2 \left(1 - \frac{\gamma - 1}{2} M_\infty^2\right) \quad (8)$$

$$\rho_0 = \rho_\infty \left(1 - \frac{\gamma - 1}{2} M_\infty^2\right)^{1/(\gamma - 1)} \quad (9)$$

For convenience, units are now chosen such that, at the stagnation point ($q = 0$), the density $\rho_0 = 1$, and sound speed $c_0 = 1$. Thus

$$c^2 = 1 - \frac{\gamma - 1}{2} q^2 \quad (10)$$

$$\rho = \left(1 - \frac{\gamma - 1}{2} q^2\right)^{1/(\gamma - 1)} \quad (11)$$

$$M^2 = \frac{q^2}{1 - \frac{\gamma - 1}{2} q^2} \quad (12)$$

$$q^2 = \frac{M^2}{1 + \frac{\gamma - 1}{2} M^2} \quad (13)$$

The maximum speed for which ρ and c^2 are positive is

$$q_{max} = \left(\frac{2}{\gamma - 1} \right)^{1/2} \quad (14)$$

for $\gamma > 1$ and unbounded otherwise. The critical speed for the transition to supersonic flow is

$$q_{cr} = \left(\frac{2}{\gamma + 1} \right)^{1/2}. \quad (15)$$

for $\gamma > -1$, unbounded for $\gamma = -1$ and non-existent otherwise. The maximum Mach number is unbounded for $\gamma \geq 1$ and is given by $(2/(1 - \gamma))^{1/2}$ for $\gamma < 1$.

The irrotationality assumption

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \quad (16)$$

implies the existence of a velocity potential ϕ such that $d\phi = udx + vdy$ where u and v are the x and y velocity components, respectively. Conservation of mass,

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0, \quad (17)$$

implies the existence of a stream function, ψ , such that $d\psi = -\rho v dx + \rho u dy$ represents the mass flux across a differential line element from left to right. The relations

$$u = \frac{\partial \phi}{\partial x}, \quad v = \frac{\partial \phi}{\partial y} \quad (18)$$

$$\rho u = \frac{\partial \psi}{\partial y}, \quad \rho v = -\frac{\partial \psi}{\partial x} \quad (19)$$

follow.

Assuming a one-to-one mapping between the physical plane (x, y) and the hodograph or velocity-angle space, (q, θ) with $(u, v) = (q \cos \theta, q \sin \theta)$, one obtains equations for the variation of the stream function and velocity potential in terms of q and θ . Thus

$$d\phi = udx + vdy = q(\cos \theta dx + \sin \theta dy) \quad (20)$$

$$d\psi = -\rho v dx + \rho u dy = \rho q(-\sin \theta dx + \cos \theta dy) \quad (21)$$

or

$$dz = dx + idy = \frac{e^{i\theta}}{q} \left(d\phi + \frac{i}{\rho} d\psi \right) . \quad (22)$$

Since dz is a perfect differential, so that the line integral in the physical plane is path independent, one obtains, considering that ϕ and ψ are functions of q and θ , the equations

$$\frac{\partial\phi}{\partial\theta} = \frac{q}{\rho} \frac{\partial\psi}{\partial q}, \quad \frac{\partial\phi}{\partial q} = -\frac{(1-M^2)}{q\rho} \frac{\partial\psi}{\partial\theta} . \quad (23)$$

Elimination of ϕ leads to an equation for the stream function

$$q^2 \frac{\partial^2\psi}{\partial q^2} + q(1+M^2) \frac{\partial\psi}{\partial q} + (1-M^2) \frac{\partial^2\psi}{\partial\theta^2} = 0 . \quad (24)$$

This is termed the Chaplygin equation for the stream function. It is a separable linear second order equation whose coefficients depend only on the speed q . This equation possesses separable solutions of the form $\psi = \psi_n(q)e^{in\theta}$. In the case of the isentropic ideal gas relation, Chaplygin noted that if one writes

$$\psi = \tau^{n/2} F_n(\tau) e^{in\theta} = \psi_n(\tau) e^{in\theta} \quad (25)$$

where

$$\tau = (q/q_{max})^2 = (\gamma-1)q^2/2 \quad (26)$$

so that

$$\tau_{cr} = (\gamma-1)/(\gamma+1) \quad (27)$$

then substitution in Equation 24 yields

$$\tau(1-\tau)F_n'' + [n+1 - (a_n + b_n + 1)\tau]F_n' - a_n b_n F_n = 0 \quad (28)$$

where

$$a_n + b_n = n - \frac{1}{\gamma-1} \quad (29)$$

$$a_n b_n = -\frac{n(n+1)}{2(\gamma-1)} . \quad (30)$$

Clearly, a_n and b_n are roots of a quadratic. In addition, we adopt the convention , $a_n < b_n$. For the convenience of avoiding complex values of a_n and b_n ,

γ will be restricted to satisfy either $\gamma > 1$ or $\gamma \leq -1$. This is easily shown. a_n and b_n are roots of the equation

$$x^2 - \left(n - \frac{1}{\gamma - 1}\right)x - \frac{n(n+1)}{2(\gamma - 1)} = 0. \quad (31)$$

Let $y = 1/(\gamma - 1)$. All roots of Equation 31 will be real for every real n , if the discriminant

$$(n - y)^2 + 2n(n + 1)y \geq 0 \quad (32)$$

or

$$y^2 + n^2(2y + 1) \geq 0. \quad (33)$$

This inequality will be satisfied for all n provided $y \geq -1/2$. Thus either $\gamma > 1$ or $\gamma \leq -1$ is required in order that a_n and b_n be real for every n .

One recognizes the solutions of Equation 28 as Gauss hypergeometric functions. The solution regular at $\tau = 0$ is of particular interest to us and is given by

$$F_n(\tau) = {}_2F_1(a_n, b_n; n + 1; \tau) = \sum_{m=0}^{\infty} \frac{(a_n)_m (b_n)_m \tau^m}{(n + 1)_m m!} \quad (34)$$

in the notation of Abramowitz and Stegun with $(a)_m \equiv (a)(a+1) \cdots (a+m-1)$ [1]. For the Chaplygin gas, $\gamma = -1$, so that $a_n = n/2$ and $b_n = (n+1)/2$. Then by a quadratic transformation formula,

$${}_2F_1(n/2, (n+1)/2; n+1; \tau) = \left(\frac{2}{1 + \sqrt{1 - \tau}} \right)^n. \quad (35)$$

(See 15.3.19 of [1].) Since Equation 24 is linear, boundary value problems may be solved by appropriate linear combinations of solutions.

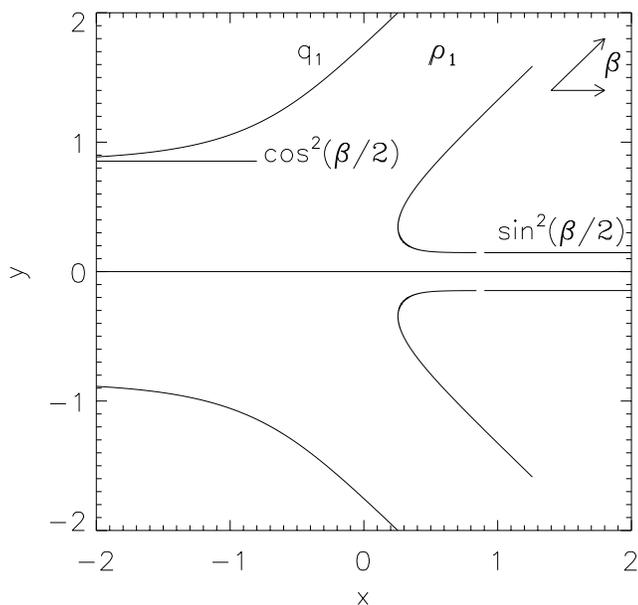


Figure 2: Plane Jet Flow

3 Chaplygin Solution to Free Surface Wall Jet Problem

Imagine a plane free surface jet of unit width impinging on a wall at an angle β and subsonic velocity $q_1 < q_{cr}$ with an incoming flux $\Delta\psi = \rho_1 q_1$ where ρ_1 is the free streamline density and q_1 is the free streamline velocity. The jet splits into two outgoing streams of asymptotic widths $(1 + \cos \beta)/2$ on the left and $(1 - \sin \beta)/2$ on the right as is required from mass and linear momentum conservation. See Figure 2.

The Chaplygin procedure takes a solution of the incompressible problem and provides a similar subsonic compressible solution. The incompressible wall jet solution for this problem can be determined by standard complex variable techniques [4, 18]. The incompressible complex potential, $W = \phi + i\psi$, is given by

$$W(\Omega) = (q_1/\pi) \left\{ \log(1 + \Omega e^{i\beta}) + \log(1 + \Omega e^{-i\beta}) \right\}$$

$$-(1 - \cos \beta) \log(1 - \Omega) - (1 + \cos \beta) \log(1 + \Omega) \} \quad (36)$$

where $\Omega = (q/q_1)e^{-i\theta}$ is the incompressible velocity in complex form. Another representation for this solution may be given by expanding each of the log functions in a Taylor series about $\Omega = 0$. Thus

$$\begin{aligned} W = & \quad -(q_1/\pi) \left\{ \sum_{n=2}^{\infty} \frac{1}{n} (q/q_1)^n e^{-in(\theta-\beta+\pi)} \right. \\ & \quad \left. + \sum_{n=2}^{\infty} \frac{1}{n} (q/q_1)^n e^{-in(\theta+\beta-\pi)} \right. \\ & \quad - (1 - \cos \beta) \sum_{n=2}^{\infty} \frac{1}{n} (q/q_1)^n e^{-in\theta} \\ & \quad \left. - (1 + \cos \beta) \sum_{n=2}^{\infty} \frac{1}{n} (q/q_1)^n e^{-in(\theta-\pi)} \right\}. \quad (37) \end{aligned}$$

The $n = 1$ terms in each series sum exactly to zero as a consequence of the required mass and momentum balance and thus do not appear. The Chaplygin procedure for writing a corresponding subsonic compressible solution from an incompressible solution is to make the correspondence

$$\left(\frac{q}{q_1} \right)^n \Rightarrow \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \quad (38)$$

in the formula for the stream function ψ where τ_1 is the value of τ on the free streamlines. Thus the stream function for compressible flow is

$$\begin{aligned} \psi = & \quad ((\rho_1 q_1)/\pi) \left\{ \sum_{n=2}^{\infty} \frac{1}{n} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \sin n(\theta - \beta + \pi) \right. \\ & \quad \left. + \sum_{n=2}^{\infty} \frac{1}{n} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \sin n(\theta + \beta - \pi) \right. \\ & \quad - (1 - \cos \beta) \sum_{n=2}^{\infty} \frac{1}{n} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \sin n\theta \\ & \quad \left. + (1 + \cos \beta) \sum_{n=2}^{\infty} \frac{1}{n} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \sin n(\theta - \pi) \right\}. \quad (39) \end{aligned}$$

An extra factor of ρ_1 is applied in the above formula since the stream function in the compressible case represents a mass flux.

The convergence theory for this series, called a Chaplygin series, has been described by Sedov [29]. A summary of the theory is given in Appendix C. The critical results give upper and lower bound solutions for $(2\tau/n)\psi'_n/\psi_n$. With natural assumptions about the shape of the isentrope ($d\rho/d\tau \leq 0$), it is shown that

$$(2\tau/n)\psi'_n/\psi_n \geq \sqrt{1 - M^2} \quad (40)$$

and

$$(2\tau/n)\psi'_n/\psi_n \leq \sqrt{(1 - M^2) + C\rho^2\tau/n^{3/2}} \quad (41)$$

where C is a positive constant which depends on the particular equation of state chosen. In Appendix C we derive another upper bound function to $(2\tau/n)\psi'_n/\psi_n$. This upper bound solution then provides an additional check on the numerical computation technique for large n . The series of Equation 39 are convergent for $0 \leq \tau < \tau_1 < \tau_{cr}$, since by integrating Equation 40

$$\frac{\psi_n(\tau)}{\psi_n(\tau_1)} \leq \exp\left(-\frac{n}{2} \int_{\tau}^{\tau_1} \sqrt{1 - M^2} d\tau/\tau\right). \quad (42)$$

For $\tau = \tau_1$ one obtains the appropriate Fourier series for the stepwise constant boundary values of the stream function. Since each term in the series is also a solution of Equation 24, Equation 39 is a valid representation for the compressible subsonic wall jet problem. Clearly, since $F_n(\tau) \rightarrow 1$, as $q_1 \rightarrow 0$ the solution reduces to the incompressible solution in the limiting case. From this representation of the stream function, all quantities of interest may be obtained.

Integration to obtain the physical plane may be accomplished in several ways since the physical plane is independent of integration path in the (q, θ) plane. As a check, two different approaches were implemented. In the first approach $\partial z/\partial q$ was evaluated for each point (q, θ) and then $z(q, \theta)$ was obtained by numerical integration with respect to q subject to $z(0, \theta) = 0$. The general term in each of the four series in Equation 39 is given by

$$\Psi_n = \frac{1}{n} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \sin n(\theta - \Theta). \quad (43)$$

where Θ is a constant depending on the particular series.

Thus utilizing Equations 23,

$$z_{nq} = \frac{\partial z_n}{\partial q} = \frac{e^{i\theta}}{\rho q} \left(-\frac{1 - M^2}{q} \frac{\partial \Psi_n}{\partial \theta} + i \frac{\partial \Psi_n}{\partial q} \right) \quad (44)$$

$$z_{nq} = \frac{e^{i\theta}}{\rho q^2} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \left(-(1 - M^2) \cos n(\theta - \Theta) + i \frac{2\tau}{n} \frac{\psi'_n(\tau)}{\psi_n(\tau)} \sin n(\theta - \Theta) \right). \quad (45)$$

Given these quantities $\partial z / \partial q$ is determined by replacing the Ψ_n in Equation 39 by z_{nq} . The values $\partial z / \partial q$ are obtained by summation and then z is determined by numerical integration with respect to q . The trapezoidal rule was exclusively used for this integration. For $\tau = \tau_1 < \tau_{cr}$, the z_q series is divergent. For $\tau = \tau_1 = \tau_{cr}$, the z_q series may be shown to be conditionally convergent by the Dirichlet test using the upper bound solution estimate of Equation 41. These facts imply that summation for points near the free surface will require the use of some type of non-linear convergence accelerator for summing the slowly convergent and divergent series.

The second technique is to integrate $\partial z / \partial \theta$ with respect to θ analytically and sum the resultant series of integrated terms. Again utilizing Equation 23

$$z_{n\theta} = \frac{\partial z_n}{\partial \theta} = \frac{e^{i\theta}}{\rho q} \left(q \frac{\partial \Psi_n}{\partial q} + i \frac{\partial \Psi_n}{\partial \theta} \right) \quad (46)$$

$$z_{n\theta} = \frac{e^{i\theta}}{\rho q} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \left(\frac{2\tau}{n} \frac{\psi'_n(\tau)}{\psi_n(\tau)} \sin n(\theta - \Theta) + i \cos n(\theta - \Theta) \right). \quad (47)$$

Integration with respect to θ leads to the particular indefinite integrals

$$z_n = \frac{e^{i\theta}}{\rho q} \frac{n}{n^2 - 1} \frac{\psi_n(\tau)}{\psi_n(\tau_1)} \times \left(-\left(\frac{1}{n} + \frac{2\tau}{n} \frac{\psi'_n(\tau)}{\psi_n(\tau)} \right) \cos n(\theta - \Theta) + i \left(1 + \frac{2\tau}{n^2} \frac{\psi'_n(\tau)}{\psi_n(\tau)} \right) \sin n(\theta - \Theta) \right). \quad (48)$$

By use of Equations 21 and 22, it is seen that the derivative of Equation 48 with respect to q is exactly z_{nq} of Equation 45. Chose $z = 0$ at $q = 0$. Since $z_n = 0$ at $q = 0$ for $n \geq 2$, z is obtained by substituting z_n for the Ψ_n in Equation 39 and summing the series. For $\tau = \tau_1 < \tau_{cr}$, the z_n series are conditionally convergent except at the singular points $\theta = \Theta$ where they are divergent. For $\tau = \tau_1 = \tau_{cr}$, the z_n series are convergent which implies that the incoming and outgoing jets become subsonic at finite points in the physical plane.

4 Evaluation of the Solution

The exact solution discussed in the previous section can be written down with relative ease. The difficulty now with this solution (as with many non-trivial exact solutions) is that the properties of the solution are not immediately obvious and an efficient and accurate numerical evaluation of the solution is needed. For purposes of verification of other more general numerical techniques to compute this solution, one would like to evaluate the exact solution with a maximum of accuracy and a minimum of effort and computer time for any chosen value of the Mach number, M , and the collapse angle, β . This turns out to require a significant effort. There are two major computational tasks: first, the Chaplygin functions, $F_n(\tau)$, must be computed, and second, the infinite series related to the solution must be effectively summed. This summation is a particular problem near the free surface since the convergence of the series is very slow. Each of these questions will be dealt with in turn.

A number of options for computing the hypergeometric functions $F_n(\tau)$ are available. The most obvious approach is to sum the hypergeometric series directly. This has the disadvantage of requiring very high precision arithmetic in order to obtain reasonable relative accuracy at large order n . This is the approach of Nieuwland who estimated for example that to obtain a relative accuracy of 10^{-10} in the computation of $\psi_{100}(.16)$ for $\gamma = 1.4$ at least 27 significant figures would be required [23]. High precision is required due to the fact that the series is alternating and the first few coefficients can be very large which results in extreme loss of significant digits. Another proposed approach is to transform the series in a way which overcomes the cancellation problem [7]. An implementation of the Miller algorithm [34] or a direct numerical solution of the differential equation might also be feasible. A general comparison of evaluation techniques was not attempted. The continued fraction algorithms described below were implemented as several desirable features were apparent at the outset.

In [6], Chaplygin used a continued fraction approximation to compute

$$(2\tau/n)\psi'_n/\psi_n = 1 + (2\tau/n)F'_n/F_n.$$

This was sufficient to allow the computation of the contraction ratio for a planar jet emanating from a slit in a semi-infinite reservoir. Frank has given a number of continued fraction representations for ratios of Gauss hypergeometric functions [11]. The representations were derived by manipulation of the three term contiguous relations for the hypergeometric function. Two of

these representations were implemented in this work and will be discussed below. Consider continued fractions of the form

$$\beta_0 + \frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2} + \frac{\alpha_3}{\beta_3} + \dots \quad (49)$$

where, for example, three terms of the continued fraction give

$$\beta_0 + \frac{\alpha_1}{\beta_1 + \frac{\alpha_2}{\beta_2}}. \quad (50)$$

The first continued fraction representation (Equation (2.5') vii of [11]) is given by the coefficients

$$\alpha_k = -\frac{(b+k)(c-a+k-1)}{(c+k-1)(c+k)}\tau, \quad k = 1, 2, 3, \dots \quad (51)$$

$$\beta_k = \frac{b-a+k}{c+k}\tau + 1, \quad k = 1, 2, 3, \dots \quad (52)$$

with $\beta_0 = 1$. This continued fraction converges to the generating function

$$\frac{F(a, b; c; \tau)}{F(a+1, b+1; c+1; \tau)} + \frac{a\tau}{c} = \frac{abF(a, b; c; \tau)}{cF'(a, b; c; \tau)} + \frac{a\tau}{c} \quad (53)$$

provided $|\tau| < 1$. The prime represents differentiation with respect to τ . The limit characteristic equation associated with the forward difference equation for the continued fraction is

$$\sigma^2 - (1 + \tau)\sigma + \tau = 0 \quad (54)$$

and has roots 1 and τ . See Appendix B. Thus for $\gamma > 1$ and subsonic values of τ ($0 < \tau < (\gamma-1)/(\gamma+1) < 1$), this continued fraction leads to an effective computation of the ratio F'/F .

The second continued fraction (Equation (2.6') ii of [11]) is given by the coefficients

$$\alpha_k = \frac{(a+k)(b+k)}{(c+k-1)(c+k)}\tau(1-\tau), \quad k = 1, 2, 3, \dots \quad (55)$$

$$\beta_k = 1 - \frac{a+b+2k+1}{c+k}\tau, \quad k = 0, 1, 2, 3, \dots \quad (56)$$

where an equivalence transformation has been applied to the coefficients given by Frank so that α_p and β_p have finite limits as p tends to infinity [5]. This continued fraction converges to the generating function

$$\frac{F(a, b; c; \tau)}{F(a+1, b+1; c+1; \tau)} = \frac{abF(a, b; c; \tau)}{cF'(a, b; c; \tau)} \quad (57)$$

provided $Re(\tau) < 1/2$. The limit characteristic equation

$$\sigma^2 - (1 - 2\tau)\sigma - \tau(1 - \tau) = 0 \quad (58)$$

has roots $1 - \tau$ and $-\tau$. The requirement that the root $1 - \tau$ be the dominant root of the forward difference equation for the continued fraction leads to the condition $Re(\tau) < 1/2$. For $\gamma \leq -1$, subsonic values of τ will lie in the range $-\infty < \tau \leq 0$, so that this second expansion provides an efficient means of evaluating F'/F for negative values of τ . This second continued fraction representation may be derived directly from the differential equation by successive differentiation.

The ratios F'/F are needed to compute shapes of free streamlines and related fundamental quantities (e.g. the compression ratio for a jet from a slit). For some applications this may be sufficient and no further information about F would be necessary. However, since the whole flow field is of interest for code verification studies, the F'/F information obtained above can be used to compute F in a useful form. Numerical integration leads immediately to values of $\log F(\tau)$ with $\log(F(0)) = 0$. The trapezoidal rule with Romberg extrapolation was used for the numerical integration scheme. The ratio $\psi_n(\tau)/\psi_n(\tau_1)$ is given by

$$\psi_n(\tau)/\psi_n(\tau_1) = \exp\{(n/2)\log(\tau/\tau_1) + \log F_n(\tau) - \log F_n(\tau_1)\}. \quad (59)$$

Forming sums and differences of logs prior to exponentiation has the advantage of avoiding underflow errors for large values of n . The above algorithm was found to be accurate, reliable and effective with no apparent numerical difficulties. Results were compared with tables of the Chaplygin functions, $F_n(\tau)$, given by Ferguson and Lighthill [10] and with the exact solution for $\gamma = -1$.

Figures 3 through 5 show the form of $F_n(\tau)$, $\psi_n(\tau)/\psi_n(\tau_1)$ and $(2\tau/n)\psi'_n/\psi_n$, respectively, for increasing order n for the standard copper equation of state described in Appendix A and with $\tau_1 = \tau_{cr} = 0.664$. This $\tau = \tau_1$ location is

shown in the graphs as a vertical line. Figure 5 shows the lower bound curve $\sqrt{1 - M^2} = \sqrt{(1 - (\tau/\tau_{cr})) / (1 - \tau)}$ which is approached for large n (Sedov, 1965) and the upper bound estimate derived in Appendix C for $n = 200$. Figure 6 shows the number of continued fraction terms utilized as a function of τ and n .

Once the Chaplygin functions are available, it is necessary to sum the series containing z_{nq} and z_n . Evaluation of Equation 45 and 48 is straightforward for all values of q if it is observed that

$$\lim_{q \rightarrow 0} \frac{\psi_n(\tau)}{q^m} = \frac{\delta_{n,2} \delta_{m,2}}{q_{max}^2}, \quad n \geq 2, \quad m = 1, 2. \quad (60)$$

The series solutions given by the Chaplygin technique are very slowly convergent for points near to the free surface. For $q = q_1$ but away from the singular points the z_n series are conditionally convergent and the z_q series are divergent. It therefore seems necessary to sum the series using a convergence accelerator which will successfully accelerate the convergent series as well as sum the divergent z_q series on the boundary. Both summations are necessary because the value of z_q is needed to compute velocity gradients.

What is meant by the "sum" of a divergent series? A series can be thought of as a limited representation of an underlying function. This representation makes mathematical sense only where it is convergent. However, it can be meaningfully related to an extension of this function outside the original domain of validity of the representation. For example, the complex series $1 + z + z^2 + \dots$ is convergent only for $|z| < 1$ while the equivalent representation $1/(1 - z)$ is valid everywhere except at the pole $z = 1$. Successful series acceleration and summation techniques essentially extract a more fundamental representation from a sequence of finite sums.

The algorithms tried were the ϵ -algorithm, the θ -algorithm and the Levin-t and Levin-u algorithms. Theorems on the regularity and accelerative properties of these various algorithms for various sequences are given by Wimp [33]. The accelerators were applied in turn to each of the four sums in the series representation for $\partial z / \partial q$ and z . The sums were computed separately and as complex sequences. This preserves the simple structure of the sums and precluded the failure of the acceleration algorithm due to the presence of zero or very small terms in the real or imaginary parts. Only the ϵ -algorithm was found to be successful in diagonal modes (algorithm order increasing with number of terms). The other three algorithms appeared to have a great deal

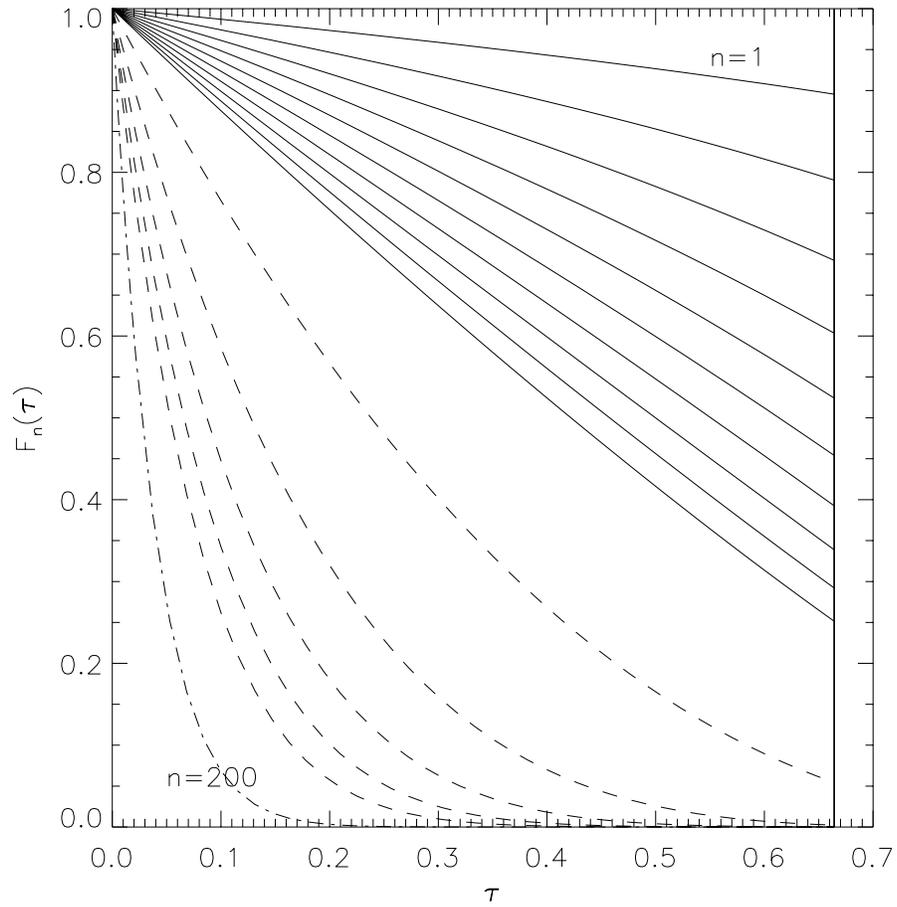


Figure 3: The function $F_n(\tau)$ for $M = 1$ for the standard Cu isentrope. Solid - $n = 1$ to 10 by 1; dashed - $n = 20$ to 100 by 20; dash-dot - $n = 200$.

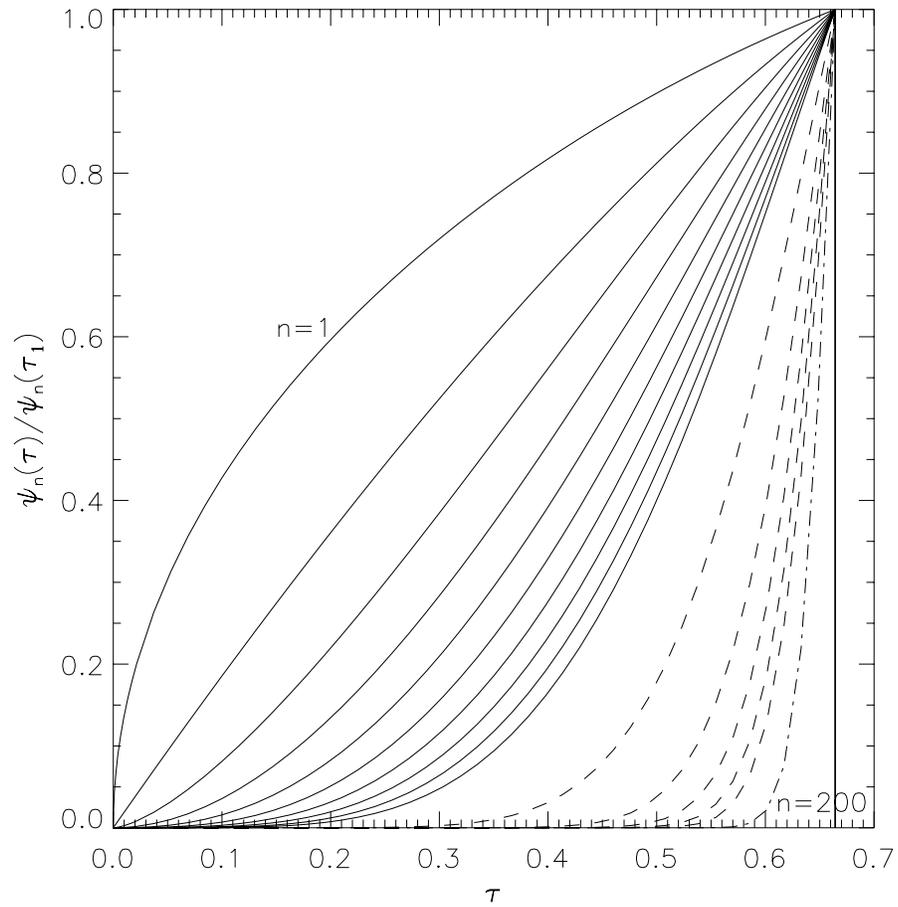


Figure 4: The function $\psi_n(\tau)/\psi_n(\tau_1)$ for $M = 1$ for the standard Cu isentrope. Solid - $n = 1$ to 10 by 1; dashed - $n = 20$ to 100 by 20; dash-dot - $n = 200$.

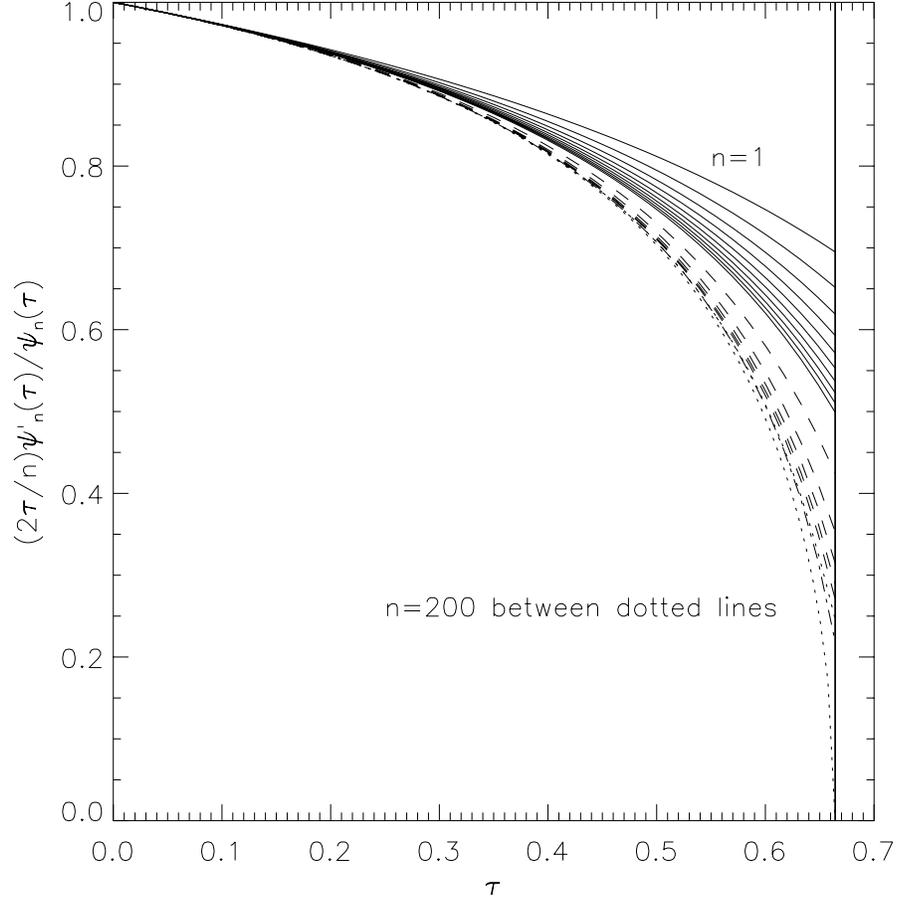


Figure 5: The function $(2\tau/n)\psi'_n/\psi_n$ for $M = 1$ for the standard Cu isentrope. Solid - $n = 1$ to 10 by 1 ; dashed - $n = 20$ to 100 by 20 ; dash-dot - $n=200$. Also shown is the lower bound solution $(\sqrt{1 - M^2})$ and the upper bound solution for $n = 200$ in dotted lines.

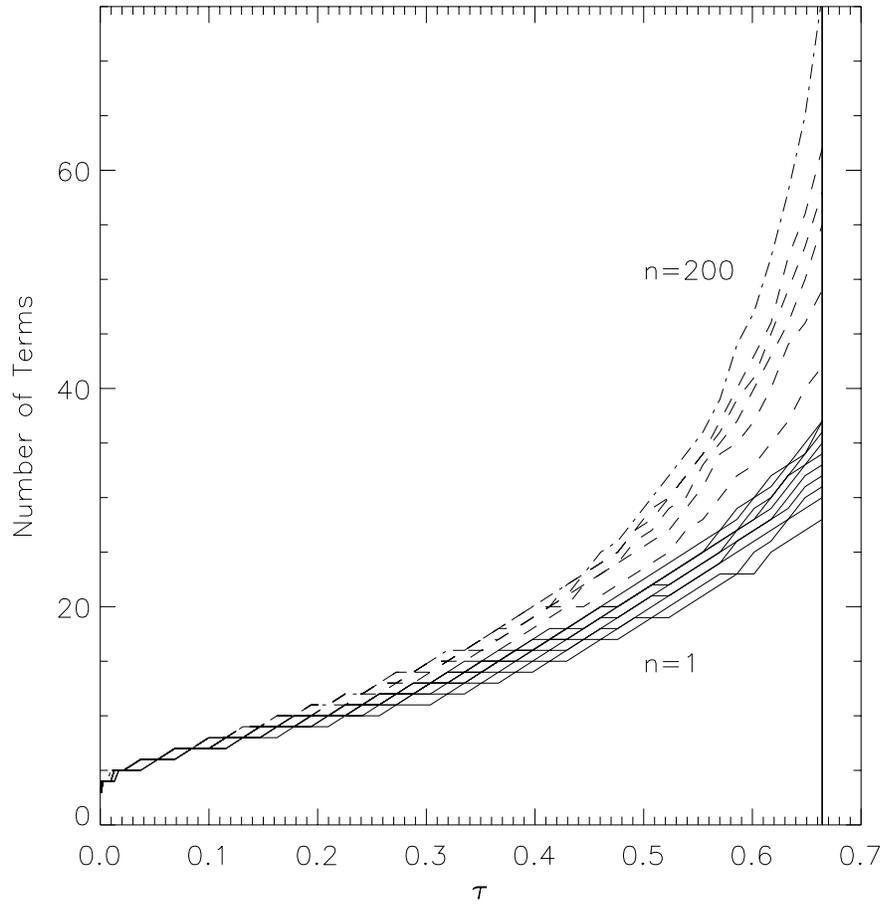


Figure 6: Numbers of terms in continued fraction as a function of n for $M = 1$ and the standard Cu isentrope. Solid - $n = 1$ to 10 by 1; dashed - $n = 20$ to 100 by 20; dash-dot - $n = 200$.

of difficulty for parameter values near and on the free surface. An extensive study of the details of this apparent failure has not been attempted. However, it appears reasonable that the ϵ algorithm succeed for the series discussed here since the algorithm will successfully compute the analytic continuation of meromorphic functions in the complex plane with a finite number of poles. See page 131 of [33].

The ϵ -algorithm has been implemented previously by Nieuwland (1967) to accelerate the convergence of Chaplygin series [23]. The ϵ -algorithm is an economical procedure for evaluating the Schmidt transformation or iterated Shank's transformation for accelerating the convergence of certain sequences. Given a sequence $s_m, m \geq 0$ with m integral the ϵ -algorithm is defined by

$$\epsilon_{k+1}^{(m)} = \epsilon_{k-1}^{(m+1)} + (\epsilon_k^{(m+1)} - \epsilon_k^{(m)})^{-1}, \quad m, k \geq 0 \quad (61)$$

$$\epsilon_{-1}^{(m)} = 0, \quad \epsilon_0^{(m)} = s_m, \quad m \geq 0. \quad (62)$$

The values $\epsilon_{2k}^{(m)}$ are used as estimates for the limit of the sequence s_m given by the partial sum of the series. An outline of the other unsuccessful algorithms is given in Appendix D.

5 Properties of the Jet Solutions

The solution outlined in the previous section varies with the incoming Mach number, the angle β and the equation of state parameter γ . This is a large parameter space. However, an attempt will be made to illustrate the various changes in solution characteristics. The basic equation of state values used here correspond to copper and are given in cgs-ev units in Appendix A. The length scale for the problem is set by the incoming jet width and is assumed to be 1 centimeter.

The variation of the pressure, density and energy are all computable from the velocity, q . Additional kinematic quantities are also of great interest and are easily accessible during the course of the solution. In particular, the velocities $u = q \cos \theta$ and $v = q \sin \theta$ along with the gradients $u_x = \partial u / \partial x$, $u_y = \partial u / \partial y = \partial v / \partial x = v_x$, and $v_y = \partial v / \partial y$ are important variables. Three additional measures of the velocity may also be instructive. These are the rate of expansion or dilatation, $u_x + v_y$, the norm of the rate-of-strain tensor, $|\mathbf{D}|$, and the norm of the deviatoric rate-of-strain tensor, $|\mathbf{D} - \frac{1}{3}(\text{tr}\mathbf{D})\mathbf{I}|$. This last quantity is of particular interest since, in the case of plastic deformation, it can be considered a measure of the rate of plastic work. The solution described here does not include, of course, plastic work, but it is thought that the metric may be useful when comparing with heating rates for jets in which the fluid motion is not highly perturbed by plasticity.

The gradients of velocity, (u, v) with respect to (x, y) must be computed from the derivatives of (x, y) with respect to (q, θ) which are known from the hodograph solution technique. The inversion formulas are given below.

Differentiating the equation $x = x(q, \theta)$ and $y = y(q, \theta)$ with respect to x and y yields

$$\mathbf{I} = \begin{pmatrix} x_q & x_\theta \\ y_q & y_\theta \end{pmatrix} \begin{pmatrix} q_x & q_y \\ \theta_x & \theta_y \end{pmatrix} \quad (63)$$

where subscripts represent differentiation. Inversion then gives

$$\begin{pmatrix} q_x & q_y \\ \theta_x & \theta_y \end{pmatrix} = \begin{pmatrix} y_\theta & -x_\theta \\ -y_q & x_q \end{pmatrix} / (x_q y_\theta - y_q x_\theta). \quad (64)$$

Now since $u = q \cos \theta$ and $v = q \sin \theta$, one obtains after differentiation, substitution and some manipulation

$$u_x = (\alpha_1 + \alpha_2 \cos 2\theta - \alpha_3 \sin 2\theta) / \Delta \quad (65)$$

$$u_y = v_x = (\alpha_3 \cos 2\theta + \alpha_2 \sin 2\theta)/\Delta \quad (66)$$

$$v_y = (\alpha_1 - \alpha_2 \cos 2\theta + \alpha_3 \sin 2\theta)/\Delta \quad (67)$$

where $\alpha_1 = M^2 F_q/2$, $\alpha_2 = (2 - M^2)F_q/2$, $\alpha_3 = (1 - M^2)G_q$, and $\Delta = F_q^2 + (1 - M^2)G_q^2$. The real quantities F_q and G_q are defined by the complex valued equation

$$F_q + iG_q \equiv e^{-i\theta}(x_q + iy_q) \quad (68)$$

Equations 23 have been used to write the x_θ and y_θ terms as functions of F_q and G_q . Note that at each evaluation point, z_q can be easily computed in the course of the integration to obtain the physical plane. Three important additional measures of strain rate are the rate of expansion or dilatation

$$\dot{\nu}/\nu = -\dot{\rho}/\rho = u_x + v_y = 2\alpha_1/\Delta, \quad (69)$$

the norm of the rate-of-strain tensor, \mathbf{D} , which due to irrotationality is the same as the velocity gradient tensor,

$$|\mathbf{D}| = |\nabla(u, v)| = \sqrt{u_x^2 + v_y^2 + 2u_y^2} = \sqrt{2(\alpha_1^2 + \alpha_2^2 + \alpha_3^2)}/\Delta, \quad (70)$$

and finally the norm of the deviatoric rate-of-strain tensor

$$|\mathbf{D} - \frac{1}{3}(\text{tr } \mathbf{D})\mathbf{I}| = \left(|\mathbf{D}|^2 - \frac{1}{3}\left(\frac{\dot{\rho}}{\rho}\right)^2 \right)^{1/2}. \quad (71)$$

A stretched evaluation mesh is utilized in both the θ direction and the q direction. The θ mapping is two cubic polynomials connecting the θ values corresponding to the singularities. The θ values match at the singularities, but the slopes with respect to the linear θ are set zero at the singular values of θ . This has the effect of concentrating more points near singularities so that better coverage is obtained in physical space. The q mesh is a linear q mesh near the origin which switches to a linear τ mesh at a specified mesh number. This has the effect of generating smooth line plots in physical space. The code given in Appendix E allows for specification of the θ and τ stretching options.

Figure 8 through 16 show on the top and bottom plots on each page a comparison of on axis quantities for $\beta = 90$ and 45 degrees, respectively. Shown are the results for the γ values:-1.0, 1.4 and 4.956 corresponding to s values 0., .6 and 1.489, respectively. The plots appear to terminate

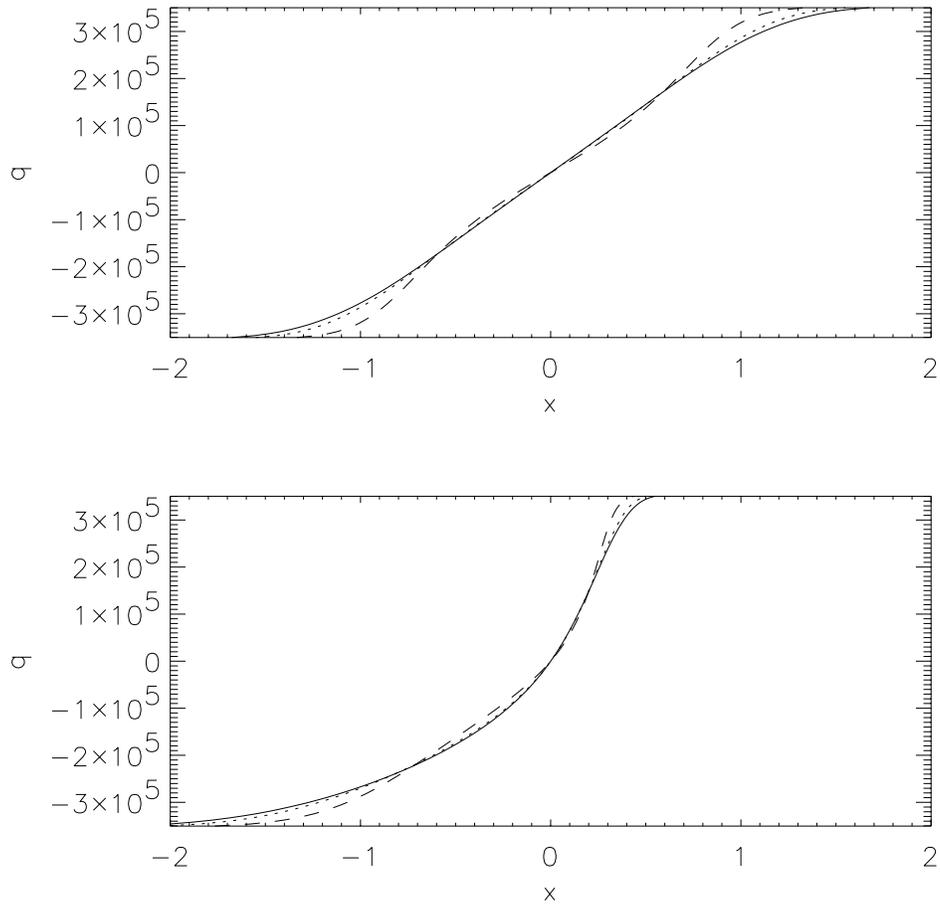


Figure 7: On axis velocity, q , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

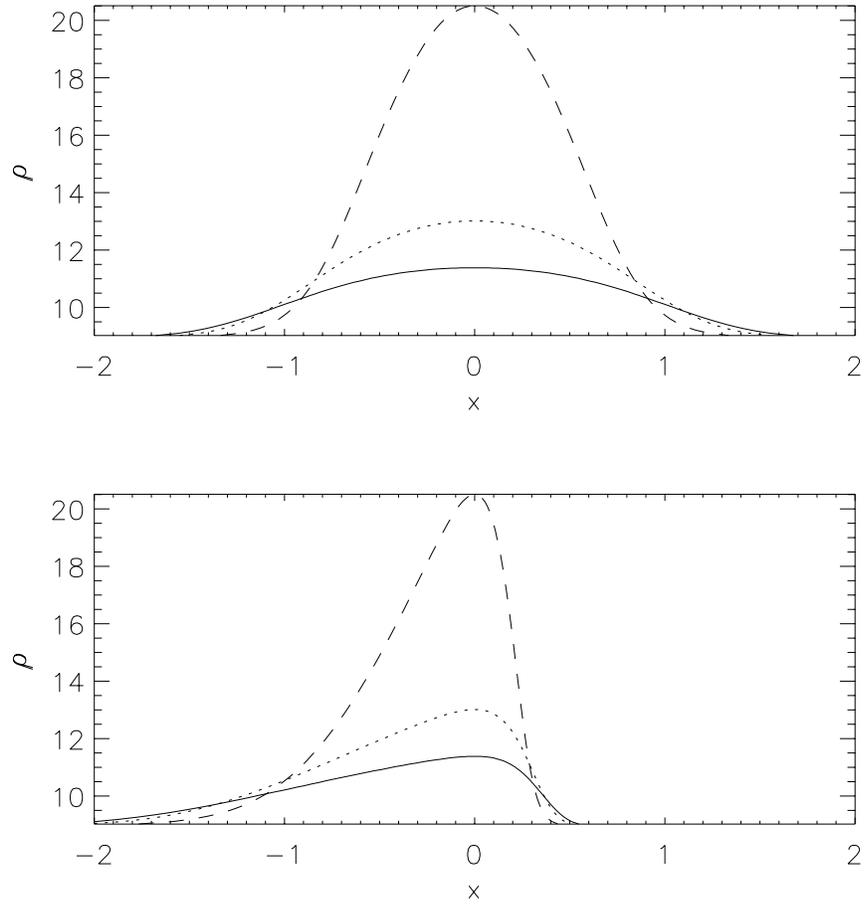


Figure 8: On axis density, ρ , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

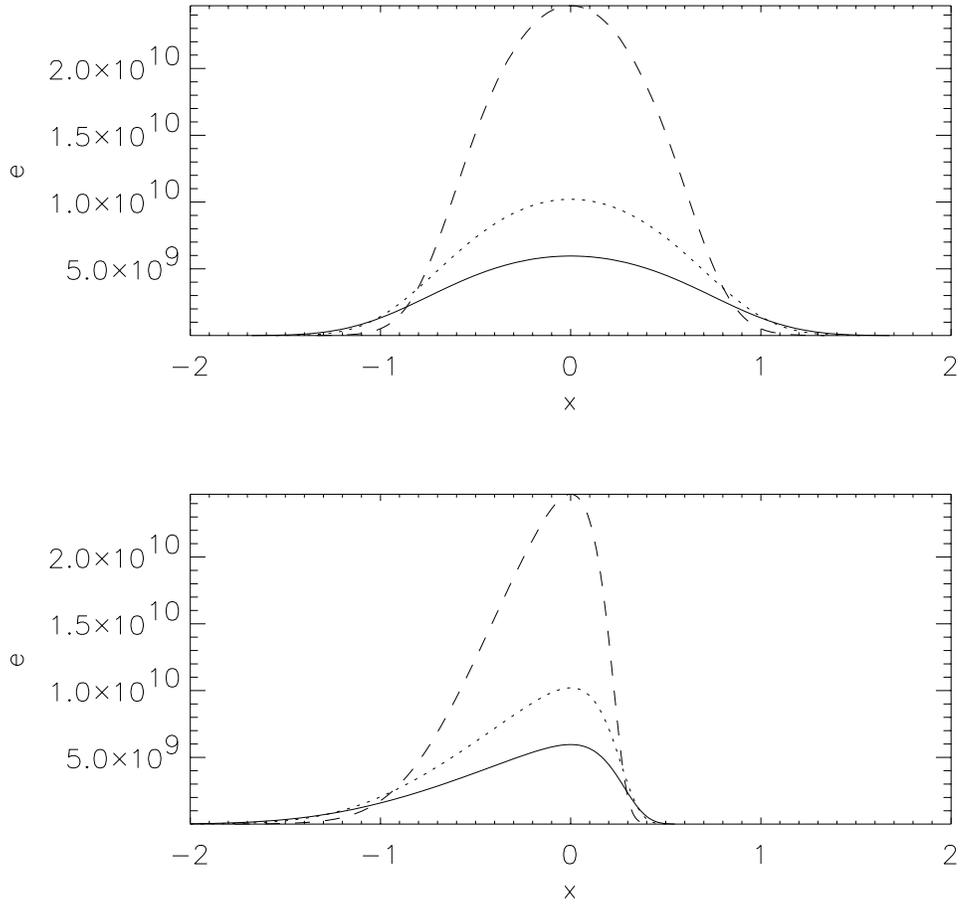


Figure 9: On axis energy, e , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

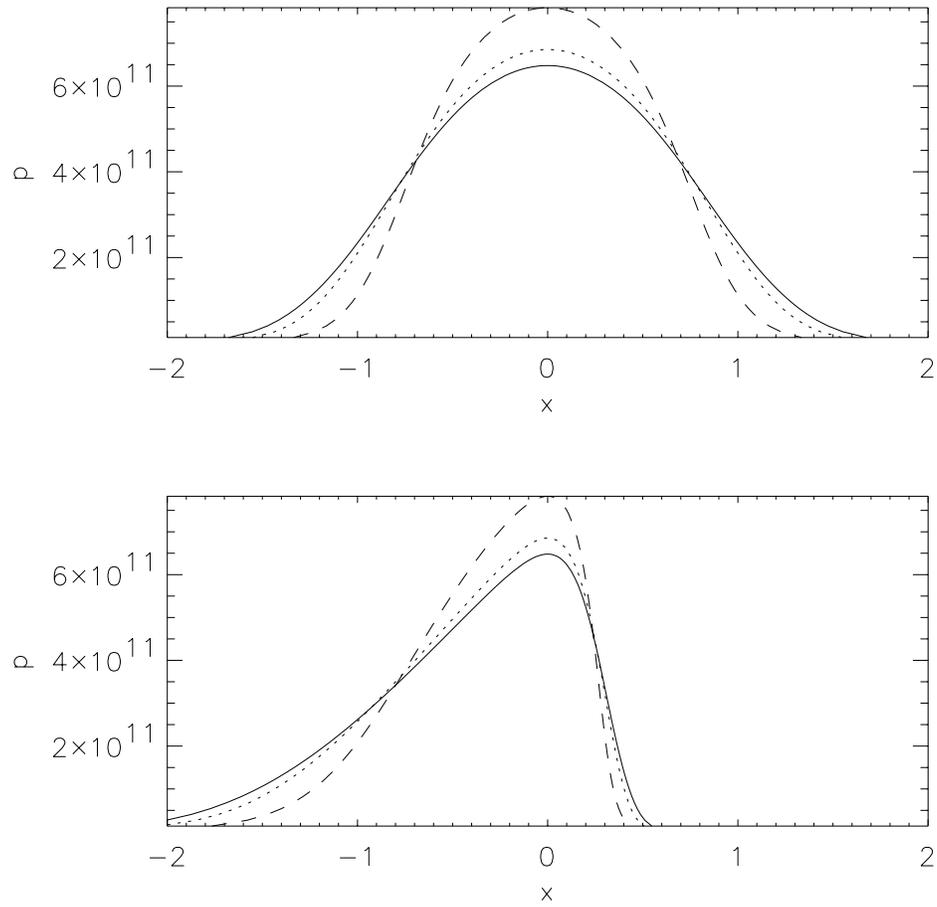


Figure 10: On axis pressure, p , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

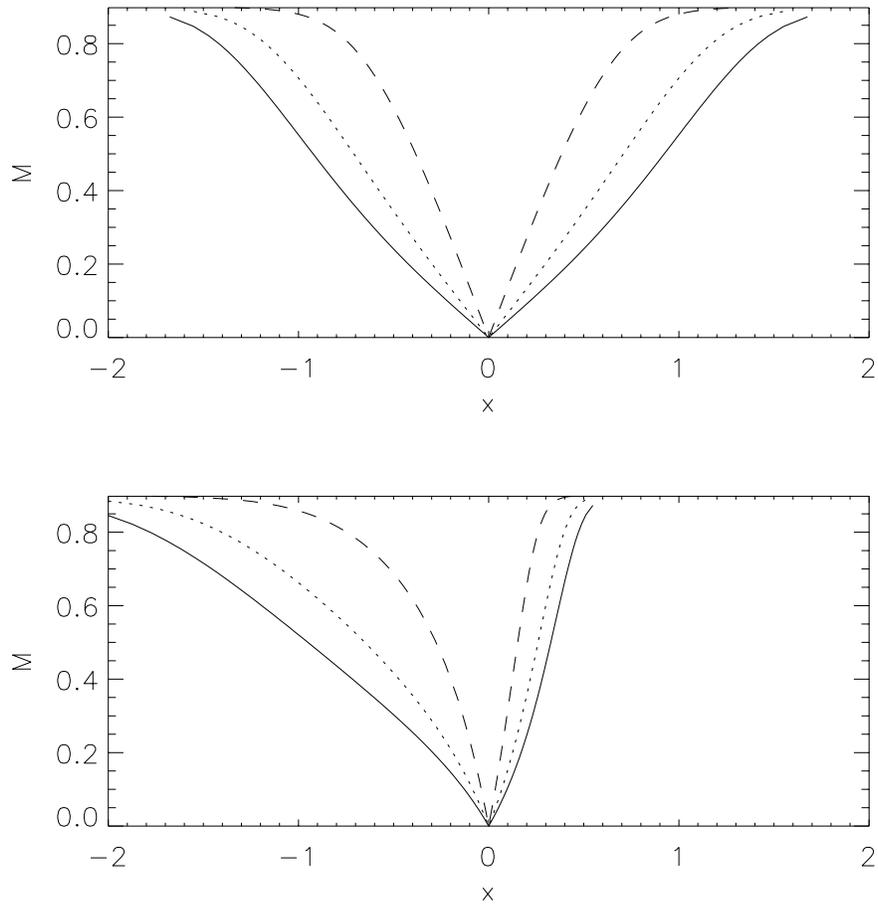


Figure 11: On axis Mach number, M , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

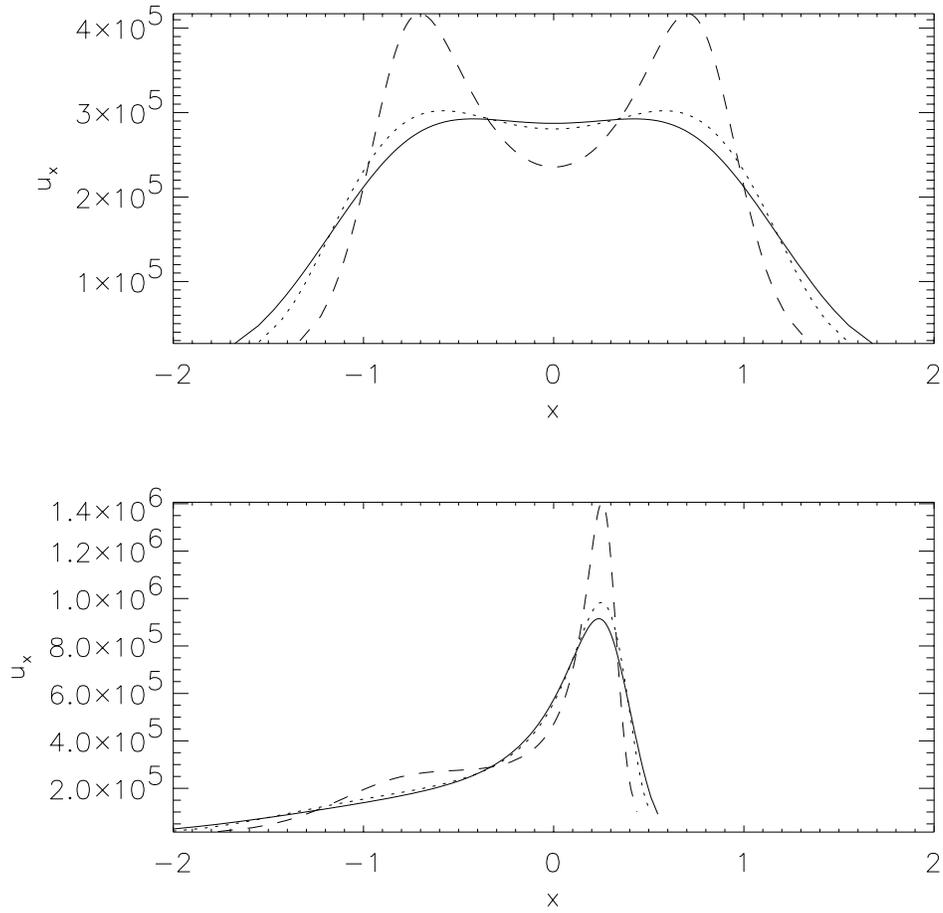


Figure 12: On axis gradient, u_x , for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

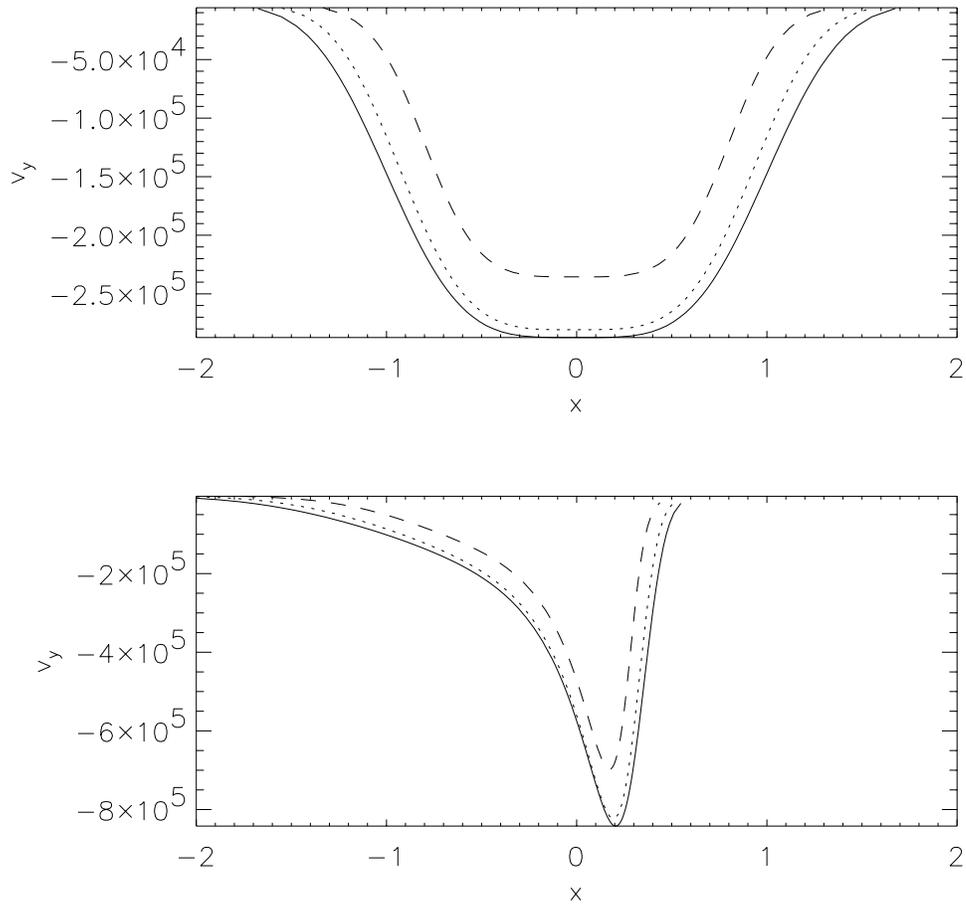


Figure 13: On axis gradient v_y for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

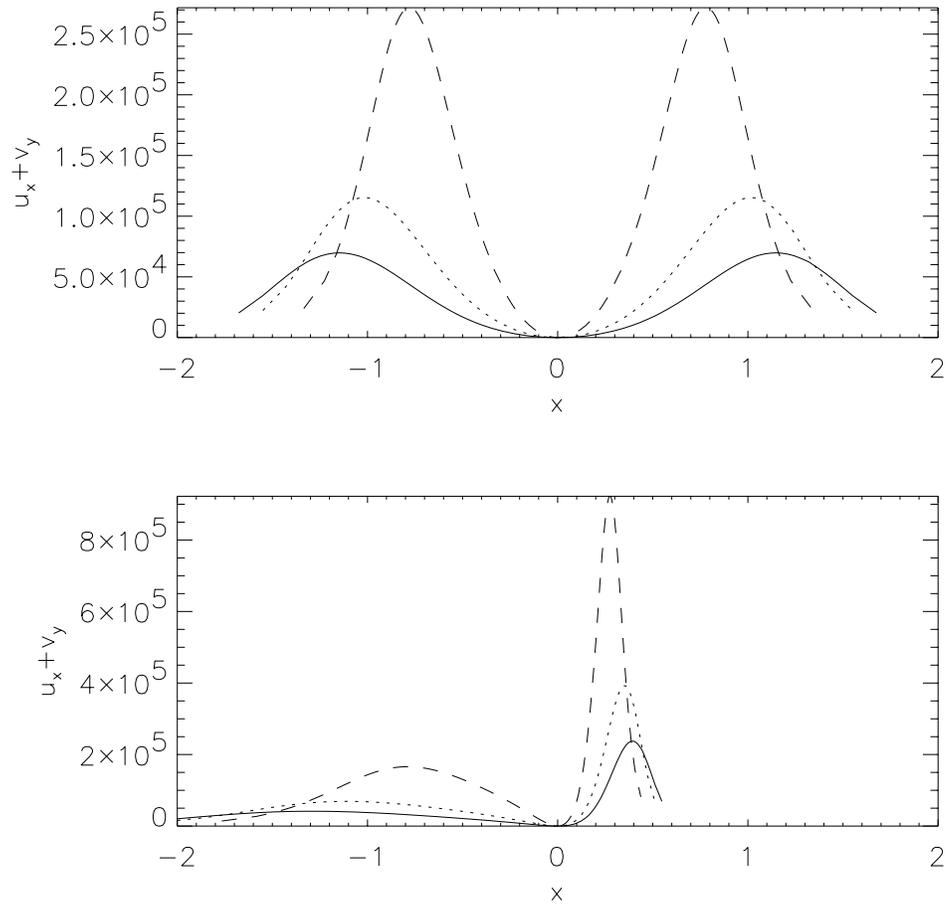


Figure 14: On axis divergence $u_x + v_y$ for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

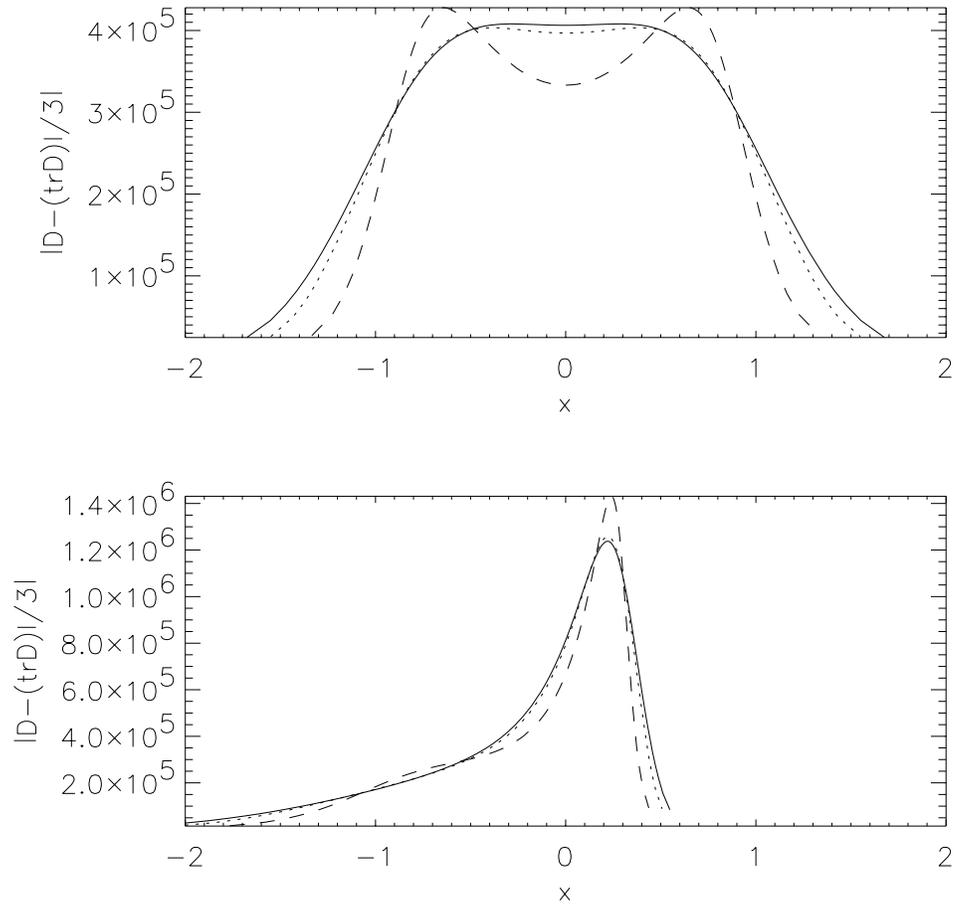


Figure 15: On axis $|\mathbf{D} - (\text{tr}\mathbf{D})\mathbf{I}/3|$ for $M_\infty = 0.9$ for γ values: -1.0 (dashed), 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

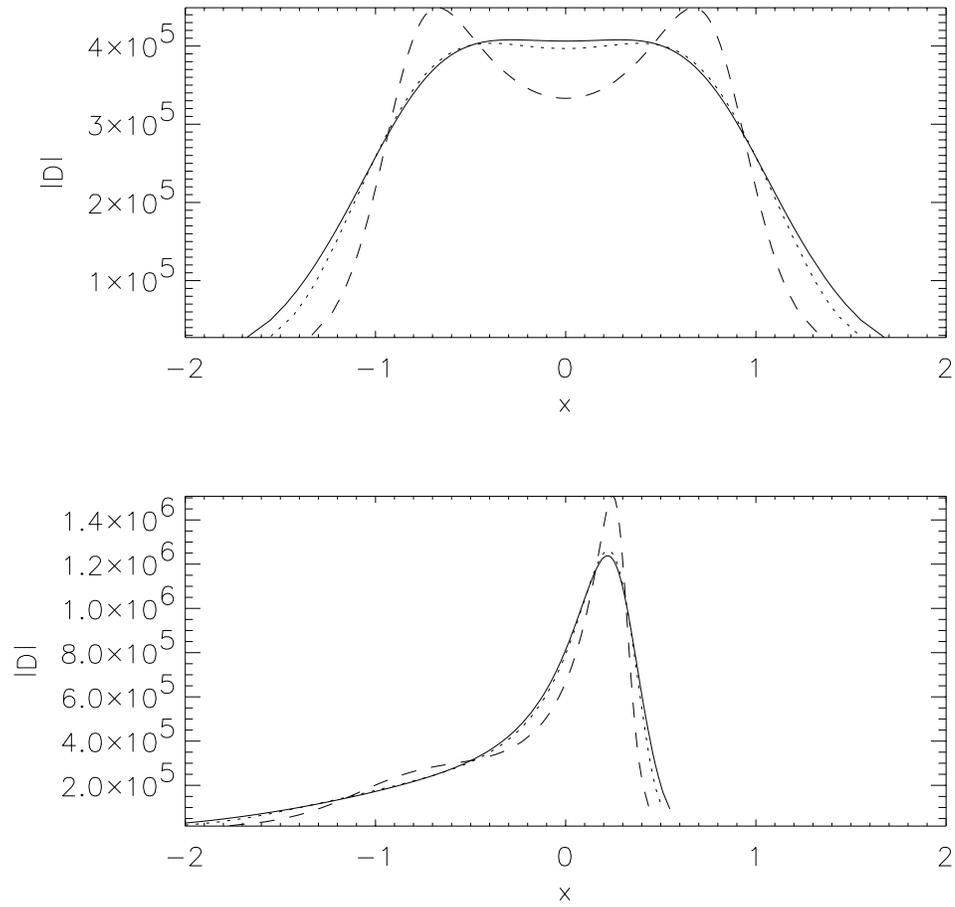


Figure 16: On axis $|\mathbf{D}|$ for $M_\infty = 0.9$ for γ values: -1.0 (dashed) , 1.4 (dotted) and 4.956 (solid) for $\beta = 90$ and 45 degrees.

prematurely in some cases. The location of the termination is a function of the finite number of velocity evaluation points.

The stiffer equations of states evidence smaller variation in the resulting flow parameters as well as a broader spatial extent for the jetting region.

Figure 17 shows the q, θ mesh in physical space. Figures 18 through 30 show contour plots of various parameters in the case of $\beta = 90$ and $\beta = 45$ degrees. Note that the largest values of the velocity gradients are at the free surface at the corner but that they are larger for the $\beta = 45$ case.

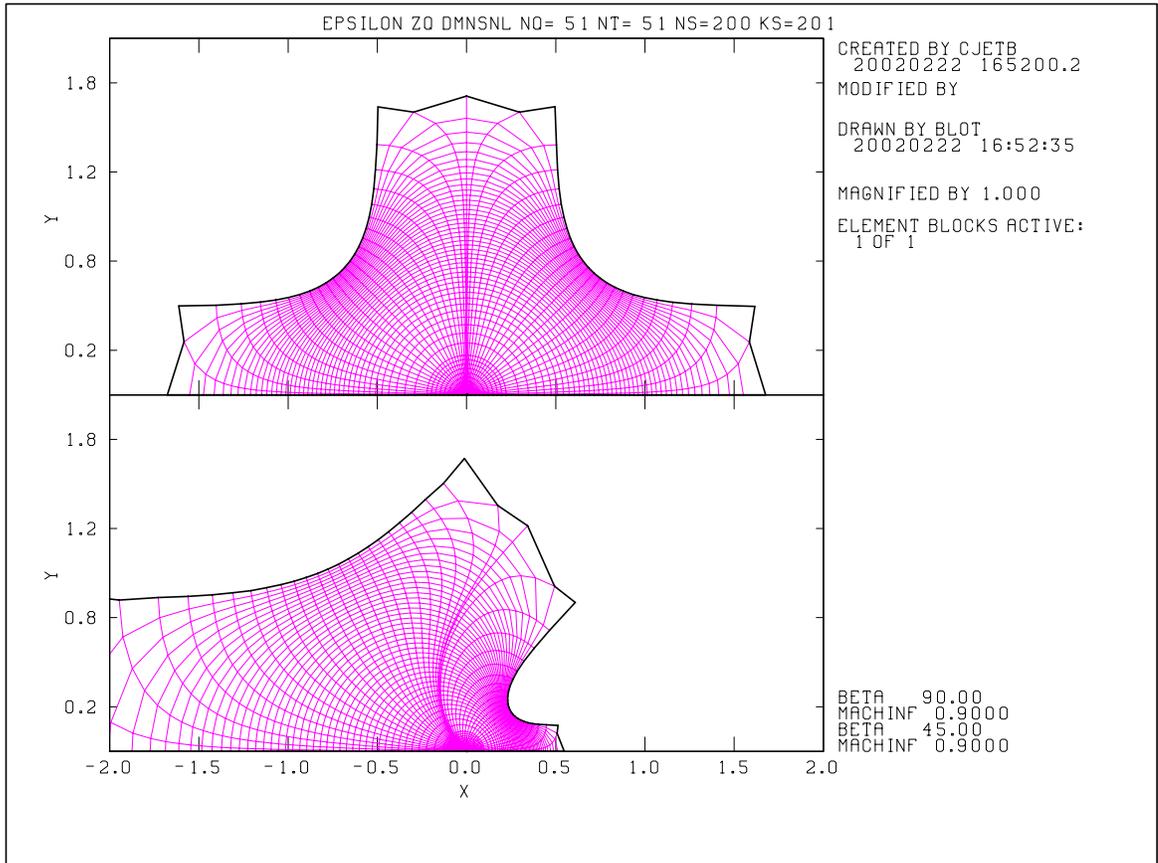


Figure 17: Evaluation mesh in the physical plane for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

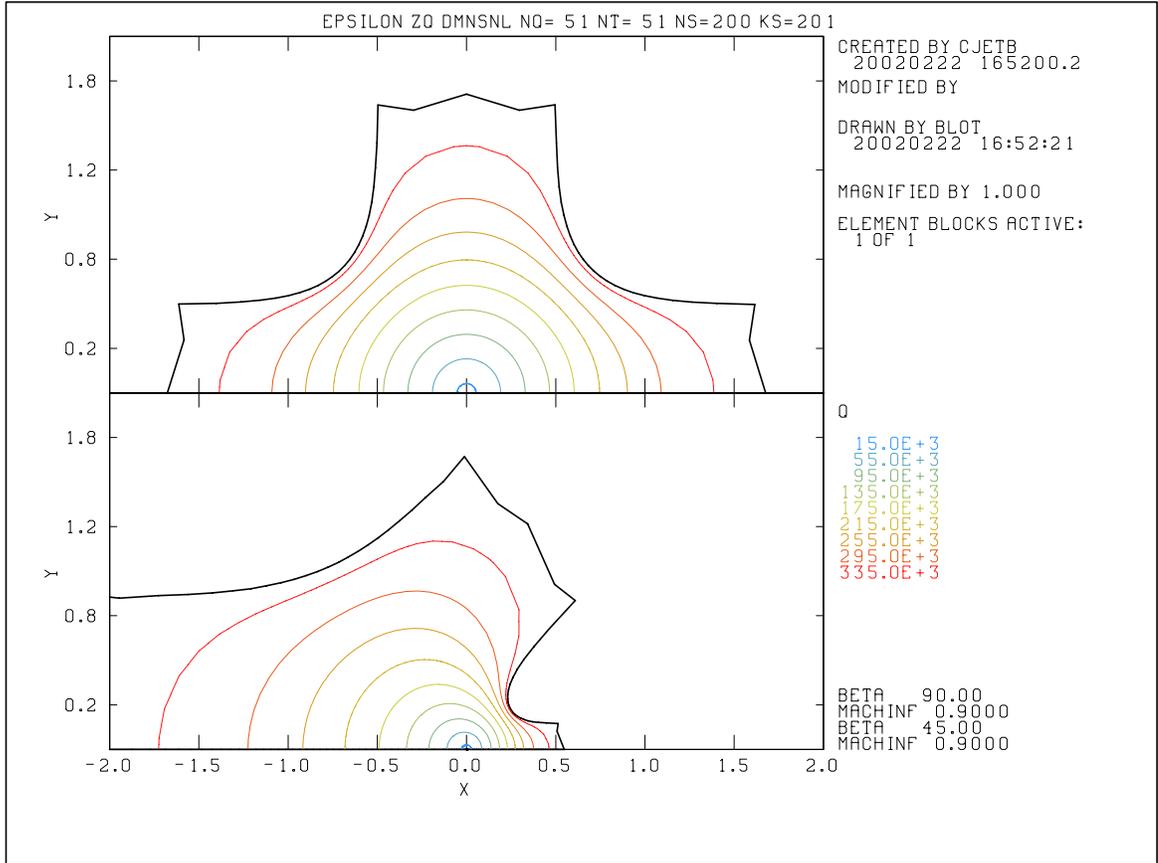


Figure 18: Velocity, q , for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

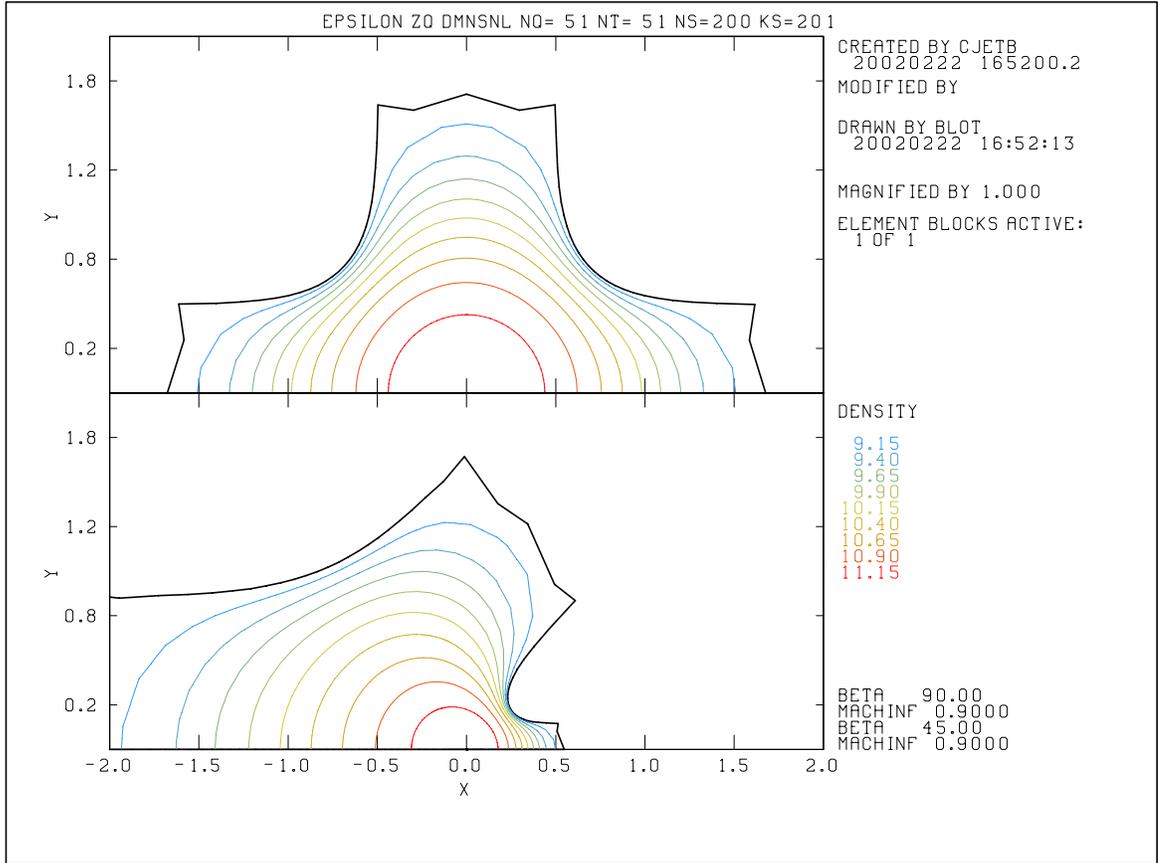


Figure 19: Density, ρ , for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

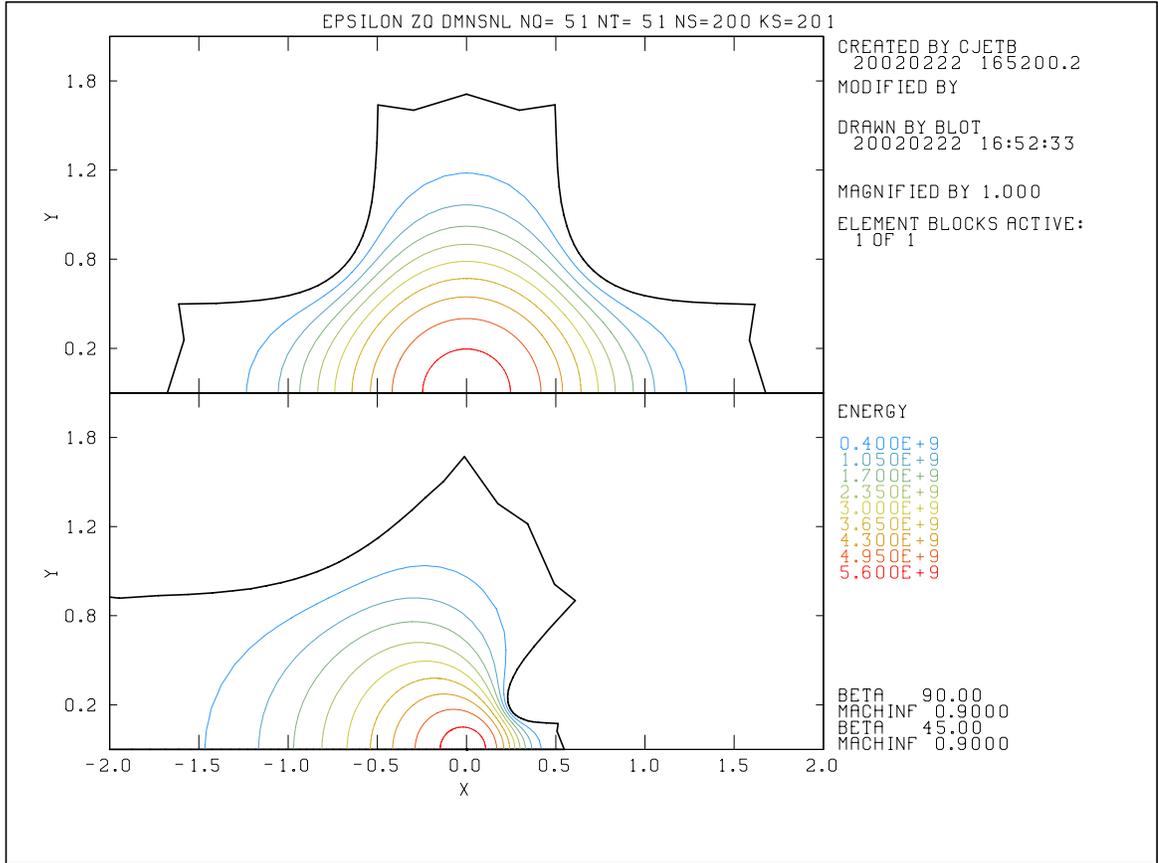


Figure 20: Internal energy, e , for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

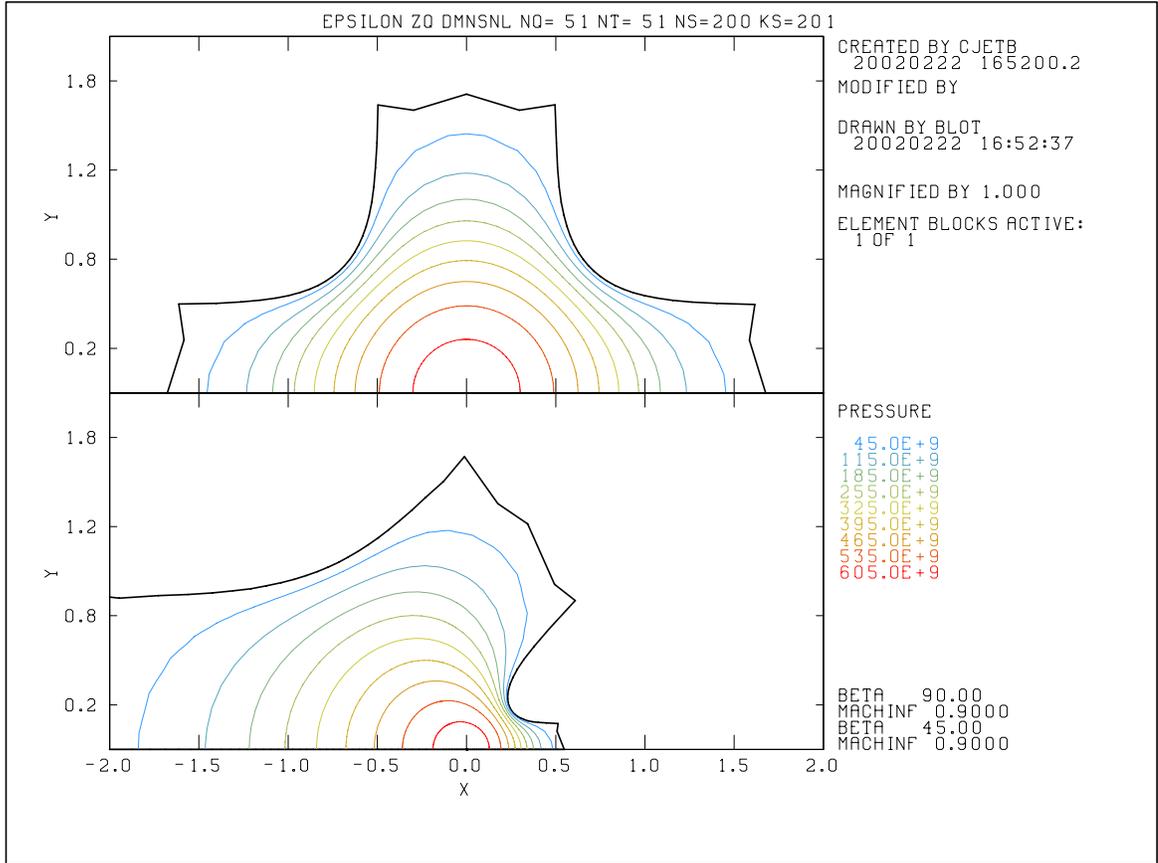


Figure 21: Pressure, p , for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

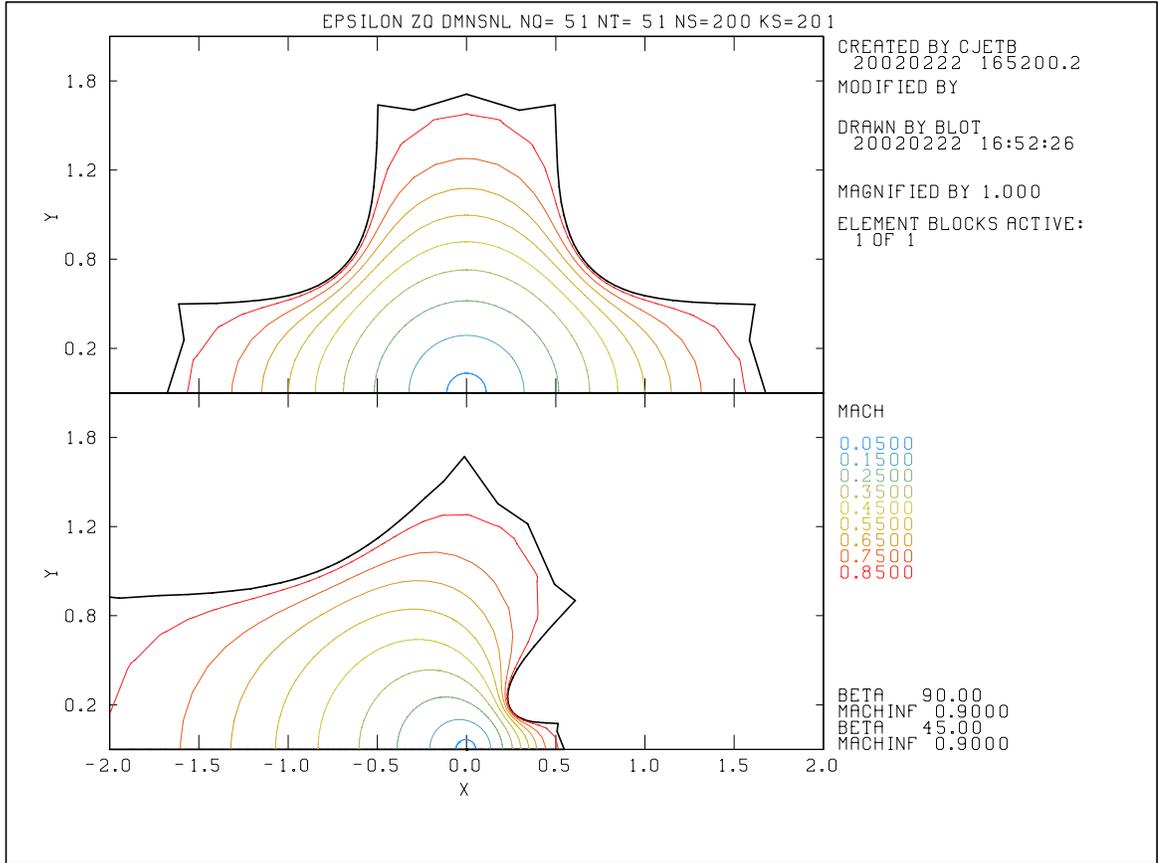


Figure 22: Mach number, M , for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

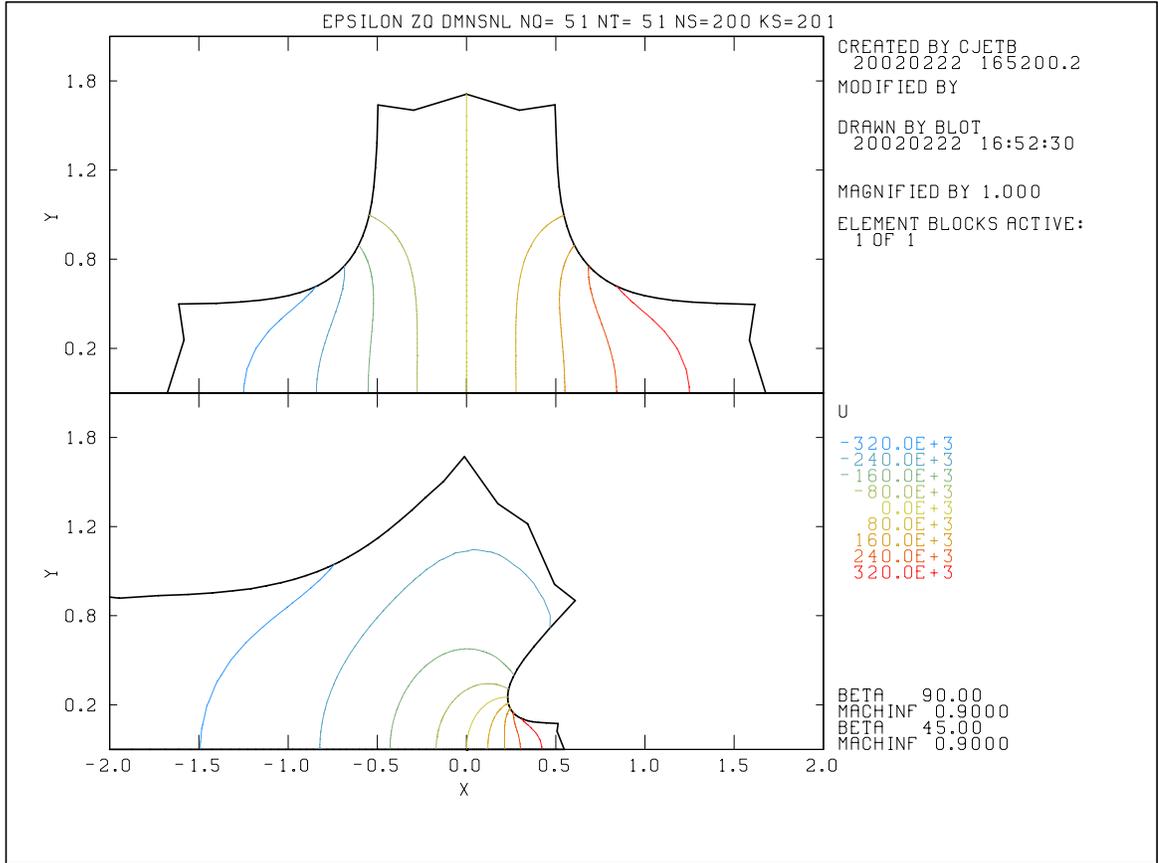


Figure 23: u for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

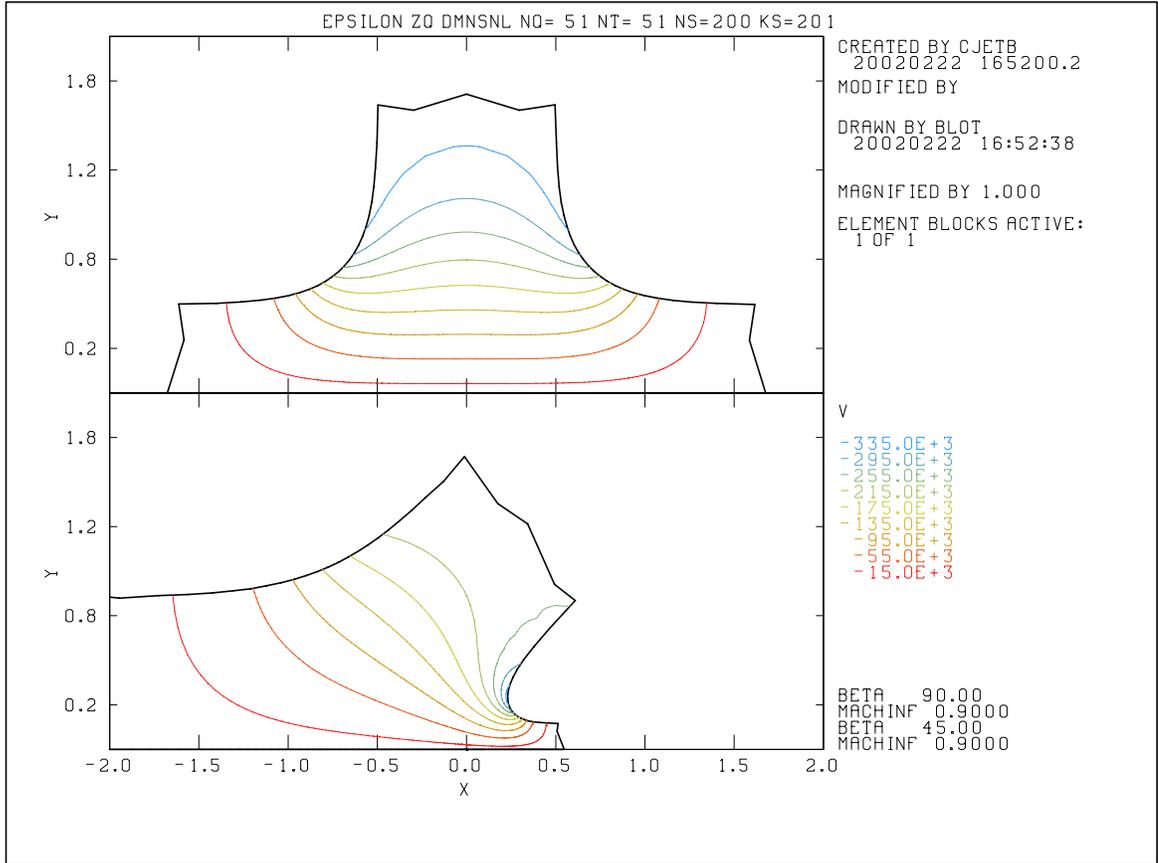


Figure 24: v for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

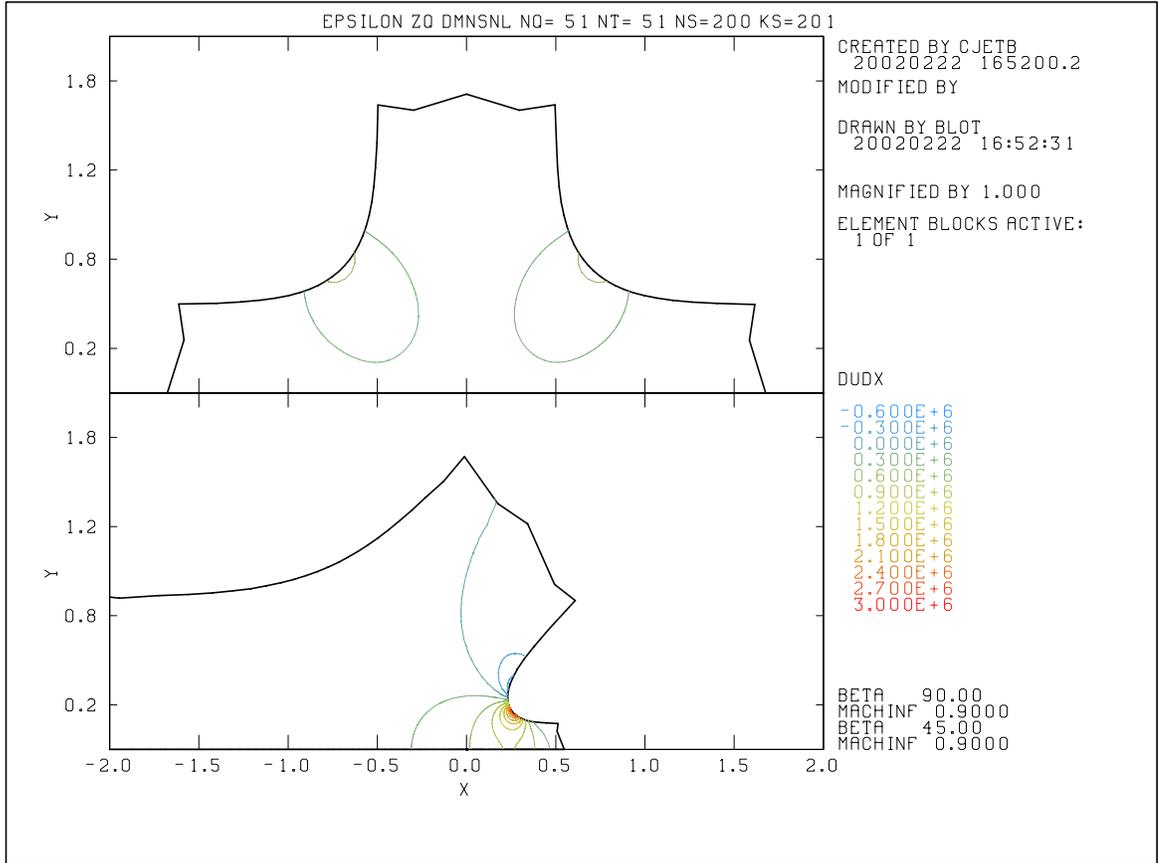


Figure 25: u_x for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

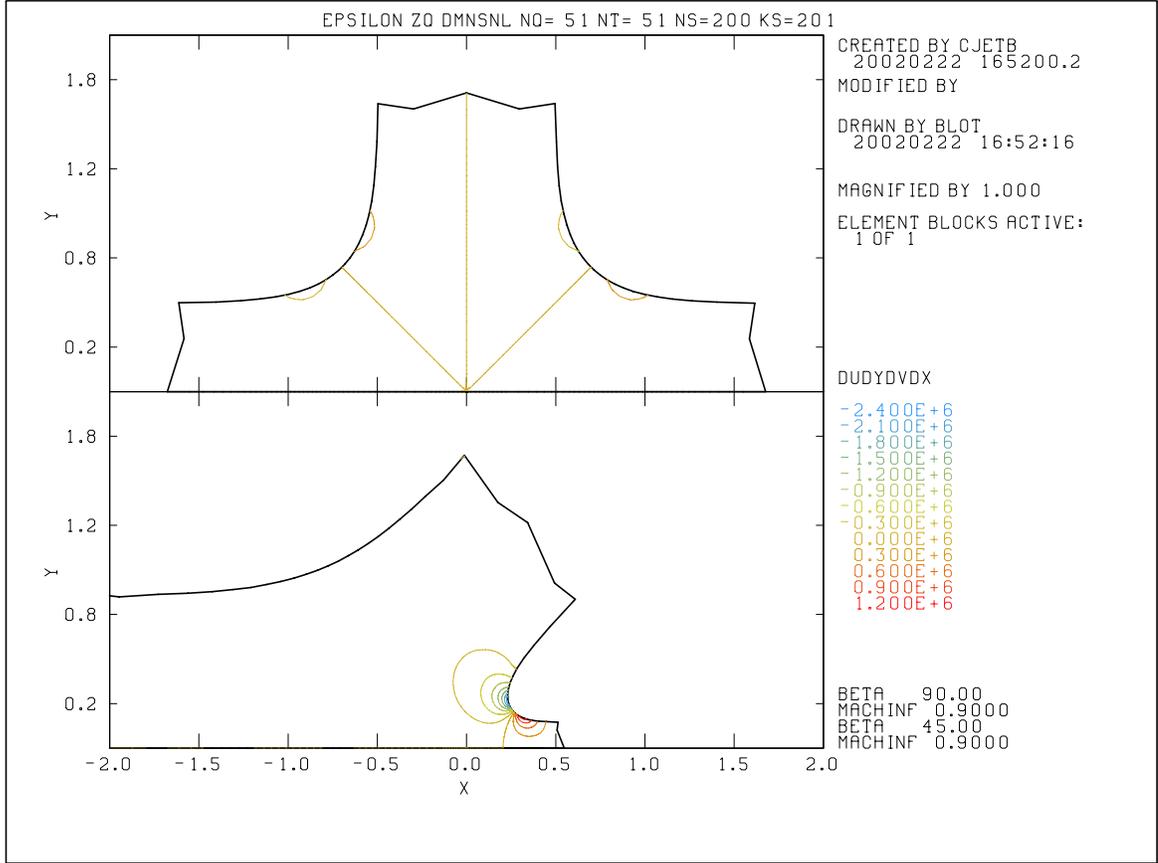


Figure 26: $u_y = v_x$ for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

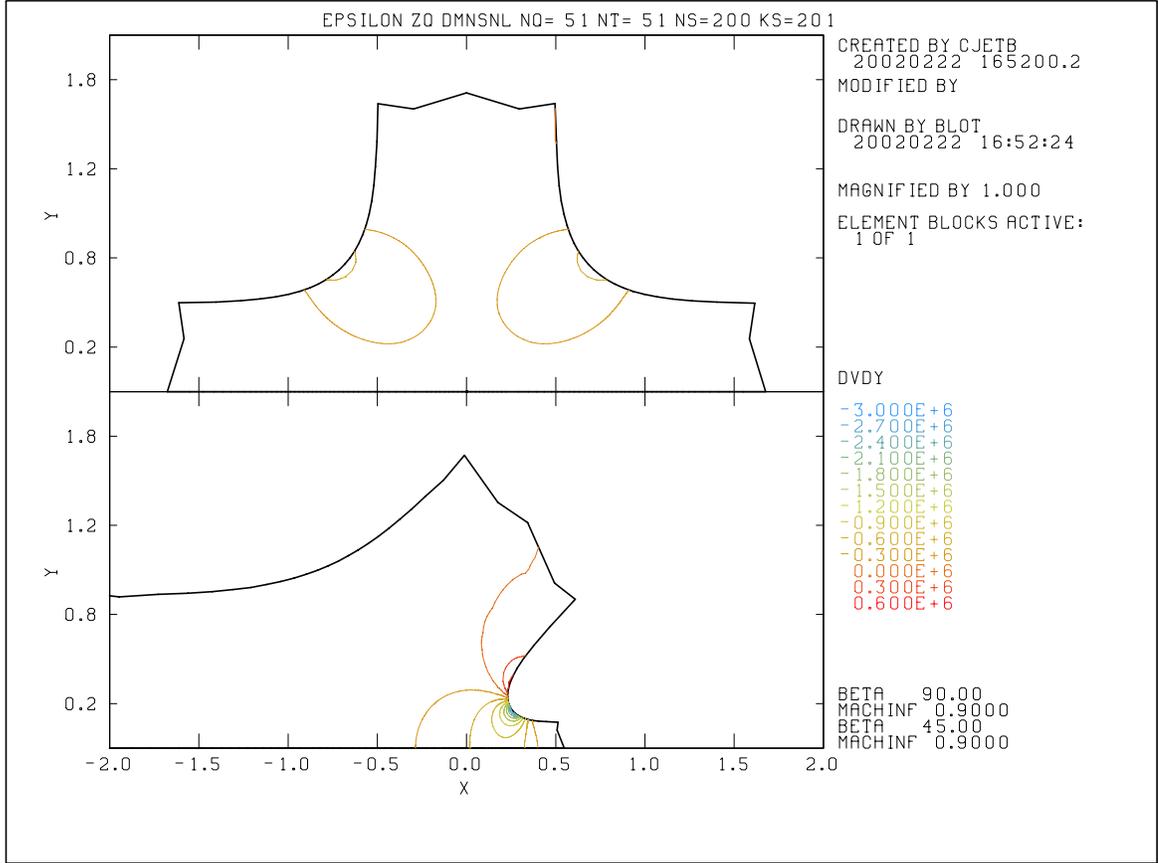


Figure 27: v_y for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

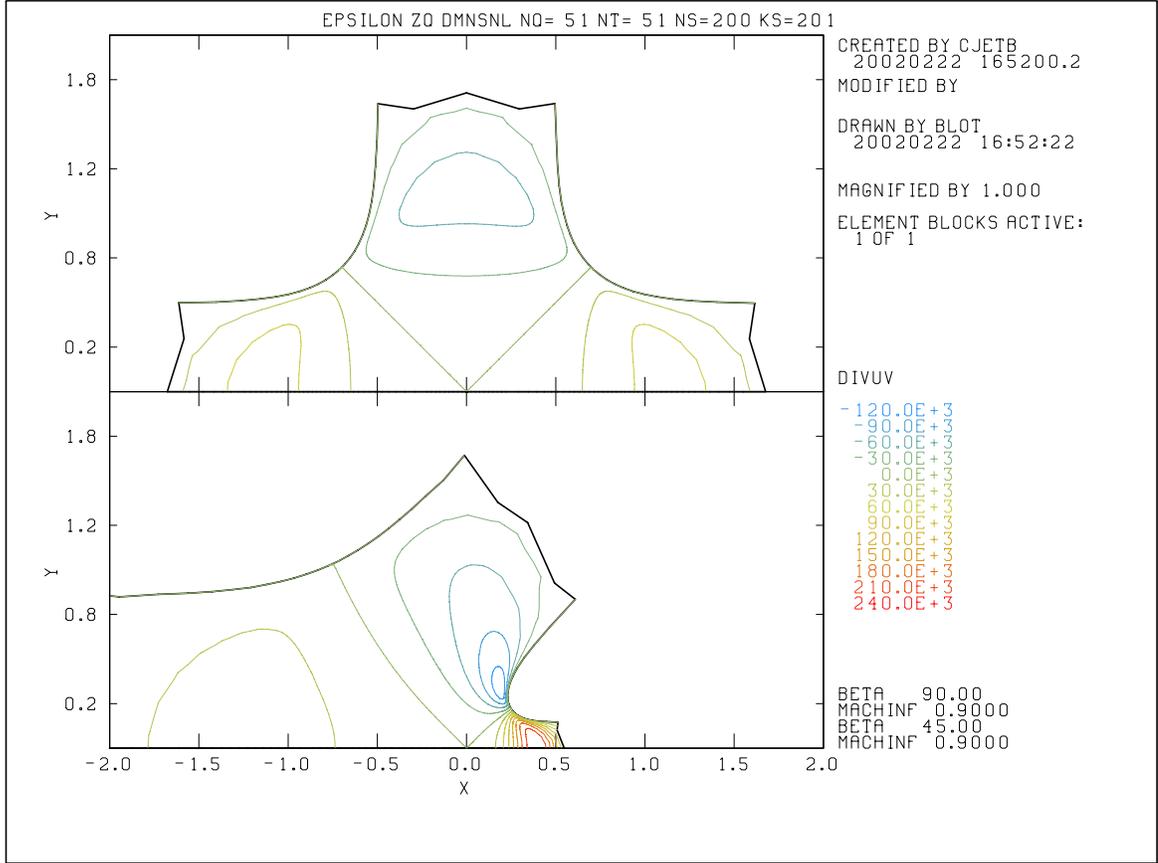


Figure 28: $u_x + v_y$ for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

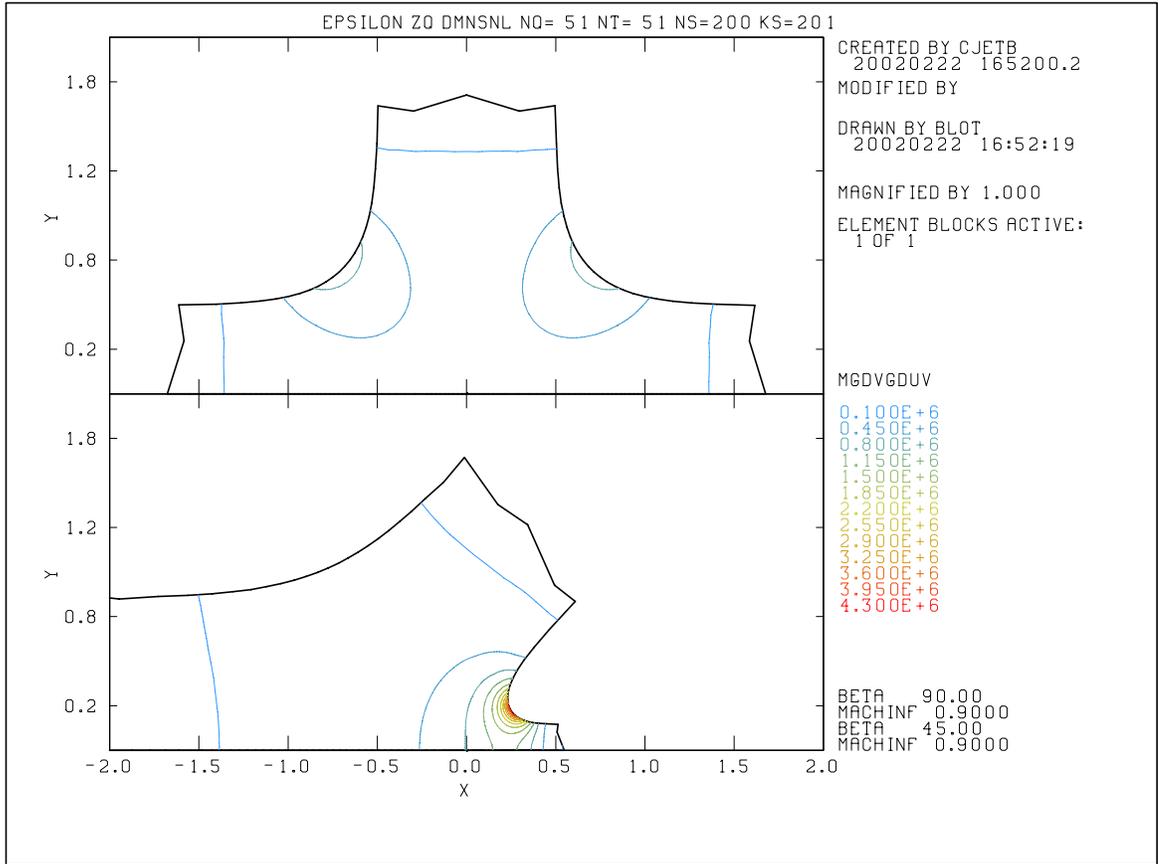


Figure 29: $|\mathbf{D} - \frac{1}{3}(\text{tr } \mathbf{D})\mathbf{I}|$ for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

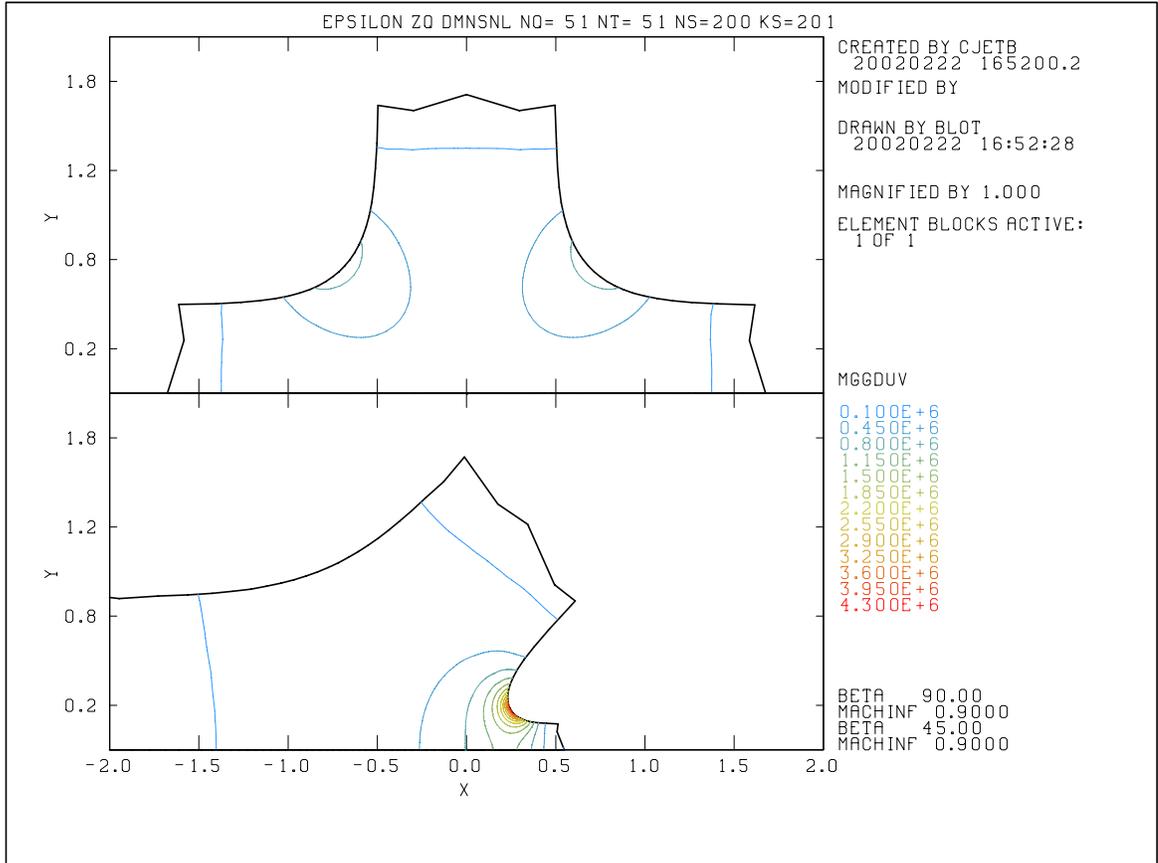


Figure 30: $|D|$ for the $\beta = 90$ and 45 degree cases for $M_\infty = 0.9$

6 Summary

An effective numerical technique for the evaluation of the Chaplygin functions $\psi_n(\tau)/\psi_n(\tau_1)$ and $(2\tau/n)\psi'_n(\tau)/\psi_n(\tau)$ without recourse to high precision floating point arithmetic has been outlined. These functions were used to compute the solution to the subsonic free-surface wall jet problem for a Murnaghan isentropic relation. In addition, formulas for various measures of strain rate have been given in terms of the hodograph variables. For particular cases detailed maps in the physical plane showing velocities, velocity gradients and several velocity gradient measures were given. These results may be used for verification comparisons with other more general purpose numerical methods.

References

- [1] M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of Mathematical Functions*, Dover Publications, Inc., New York, 1972.
- [2] L. BERS, *Mathematical Aspects of Subsonic and Transonic Gas Dynamics*, John Wiley and Sons, New York, 1958.
- [3] G. BIRKHOFF, D. P. MACDOUGALL, E. M. PUGH, AND G. I. TAYLOR, *Explosives with lined cavities*, J. Appl. Phys., 19 (1948), pp. 563–582.
- [4] G. BIRKHOFF AND E. H. ZARANTONELLO, *Jets, wakes and cavities*, in Applied Mathematics and Mechanics, vol. II, Academic Press, New York, 1957.
- [5] G. BLANCH, *Numerical evaluation of continued fractions*, SIAM Review, 4 (1964).
- [6] S. A. CHAPLYGIN, *On gas jets*, tech. rep., Moscow University, 1902. In Russian, Translated in NACA Technical Memorandum No. 1063, 1944.
- [7] G. CHIOCCHIA AND B. GABUTTI, *A new transformation for computing hypergeometric series and the exact evaluation of the transonic adiabatic flow over a smooth bump*, Computers and Fluids, 17 (1989), pp. 13–23.
- [8] P. C. CHOU, J. CARLEONE, AND R. KARPP, *Criteria for jet formation from impinging shells and plates*, J. Appl. Phys., 47 (1976).
- [9] L. P. COOK, E. NEWMAN, S. RIMBEY, AND G. SCHLEINIGER, *Sonic and subsonic axisymmetric nozzle flows*, SIAM J. Appl. Math., 59 (1999), pp. 1812–1824.
- [10] D. F. FERGUSON AND M. J. LIGHTHILL, *The hodograph transformation in transonic flow, IV. tables*, Proc. Roy. Soc, A192 (1947), pp. 135–142.
- [11] E. FRANK, *Continued Fraction Expansions for the Ratios of Hypergeometric Functions Evaluated on Computers*, vol. 419, Springer, Berlin, 1974, pp. 110–119.

- [12] A. FRIEDMAN, *Mathematics in Industrial Problems, Part 2*, vol. 24, Springer-Verlag, 1989, ch. Shaped Charge Jets and Subsonic Free Surface Flow Theory, pp. 145–155.
- [13] K. G. GUDERLEY, *The Theory of Transonic Flow*, Pergamon Press, 1962.
- [14] M. I. GUREVICH, *The Theory of Jets in an Ideal Fluid*, Pergamon Press, Oxford, 1965.
- [15] W. G. V. HOLLE AND J. J. TRIMBLE, *Temperature measurement of shocked copper plates and shaped charge jets by two-color ir radiometry*, J. Appl. Phys., 47 (1976), pp. 2391–2394.
- [16] F. JAMET, *Investigation of shaped charge jets using flash x-ray diffraction*, in Proceedings of the 8th International Symposium on Ballistics, Orlando, Florida, American Defense Preparedness Association, Oct 1984.
- [17] ———, *A flash x-ray diffraction system for shaped charge jets analysis*, in Proceedings of the 1986 Flash Radiograph Topical, Columbus, Ohio, American Society for Nondestructive Testing, 1986.
- [18] R. R. KARPP, *An exact partial solution to the compressible flow problems of jet formation and penetration in plane, steady flow*, Quart. Appl. Math, (1984), pp. 15–29.
- [19] R. R. KARPP AND S. GOLDSTEIN, *Modifications to the hydrodynamic computer code HELP*, Tech. Rep. M-4-1596, Los Alamos National Laboratory, December 1977. Work for U. S. Army Ballistic Research Lab. under Army Project Order N721-77.
- [20] R. KINSLOW, ed., *High-Velocity Impact Phenomena*, Academic Press, New York, 1970.
- [21] W. C. MILLS-CURRAN, A. P. GILKEY, AND D. P. FLANAGAN, *Exodus: A finite element file format for pre- and postprocessing*, Tech. Rep. SAND87-2997, Sandia National Laboratories, 1977.
- [22] R. V. MISES, *Mathematical Theory of Compressible Fluid Flow*, Academic Press, Inc., New York, 1958.

- [23] G. Y. NIEUWLAND, *Transonic potential flow around a family of quasi-elliptical aerofoil sections*, Tech. Rep. T-172, National Aerospace Lab NLR, Amsterdam, 1967.
- [24] W. F. NOH, *Errors for calculation of strong shocks using an artificial viscosity and an artificial heat flux*, J. Comp. Phys., 72 (1978), pp. 78–120.
- [25] C. E. PEARSON, *Extension of a numerical streamline method*, Comm. Appl. Num. Meth., 1 (1985), pp. 177–181.
- [26] E. RACAII, *Shaped charge jet heating*, Propellants, Explosives, Pyrotechnics, 13 (1988), pp. 178–182.
- [27] P. J. ROACHE, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, 1998.
- [28] J. A. SCHMITT, *Truncation error terms in the kinetic energy calculation in the HELP algorithm and their consequences*, J. Comp. Phys., 35 (1980), pp. 206–228.
- [29] L. I. SEDOV, *Two-dimensional Problems in Hydrodynamics and Aerodynamics*, Interscience Publishers, New York, 1965.
- [30] W. J. THRON AND H. WAADLAND, *Accelerating convergence of limit periodic continued fractions $k(a_n/1)$* , Numer. Math., 34 (1980), pp. 155–170.
- [31] E. D. TOMILOV, *Two-Dimensional Subsonic Gas Flows: Basic Methods and Problems*, Novosibirsk, Izdatel'stvo Nauka, 1980. Available in Russian from AIAA TIS as A81-31825.
- [32] W. P. WALTERS AND J. A. ZUKAS, *Fundamentals of Shaped Charges*, John Wiley and Sons, New York, 1989.
- [33] J. WIMP, *Sequence Transformations and Their Applications*, Academic Press, New York, 1981.
- [34] ———, *Computation with Recurrence Relations*, Pitman, Boston, 1984.

- [35] L. ZERNOW, *Recovery of shaped charge jet samples for x-ray diffraction studies*, in Proceedings from the Tenth International Symposium on Ballistics, Vol. II, American Defense Preparedness Association, October 1987.
- [36] J. A. ZUKAS, T. NICHOLAS, H. F. SWIFT, L. B. GRESZCZUK, AND D. R. CURRAN, *Impact Dynamics*, John Wiley and Sons, 1982.

A Equation of State Relations

The Murnaghan isentrope given by Equation 2 has two free parameters, κ_∞ and γ . A common form for the description of a metal Hugoniot is

$$p_H = \rho_\infty c_\infty^2 \eta / (1 - s\eta)^2 \quad (72)$$

where $\eta = 1 - \rho_\infty/\rho$. The Hugoniot and the isentrope are equal to third order in the strain. Thus one may match either of these two curves locally near $\rho = \rho_\infty$ to the same order by the Murnaghan relation. Setting the first and second derivatives with respect to $1/\rho$ of Equations 72 and 2 equal at reference conditions leads to the equations

$$\gamma \kappa_\infty = \rho_\infty c_\infty^2 \quad (73)$$

$$\gamma(\gamma + 1)\kappa_\infty = 4\rho_\infty c_\infty^2 s. \quad (74)$$

Solving for κ_∞ and γ yields the equations

$$\gamma = 4s - 1 \quad (75)$$

$$\kappa_\infty = \rho_\infty c_\infty^2 / \gamma. \quad (76)$$

The Chaplygin gas may be obtained by setting $s = 0$.

Assume a Mie-Grüneisen equation of state

$$p(e, \rho) = \bar{p}(\rho) + \rho\Gamma(e - \bar{e}(\rho)) \quad (77)$$

where \bar{e} satisfies the isentropic differential equation for the internal energy,

$$de = -pd\nu = -pd(1/\rho), \quad (78)$$

so that

$$\bar{e} = e_\infty + \frac{\kappa_\infty}{\gamma - 1} \left(\frac{1}{\rho} (\rho/\rho_\infty)^\gamma - \frac{1}{\rho_\infty} \right) + \kappa_\infty (1/\rho - 1/\rho_\infty). \quad (79)$$

The Grüneisen coefficient $\Gamma = \nu(\partial p/\partial e)_\nu$ is an arbitrary function of volume. It is common to chose $\rho\Gamma = \alpha$ constant.

In order to derive simple relations for the temperature and entropy, assume that the heat capacity at constant volume, $c_\nu = (\partial e/\partial T)_\nu$, is constant. Thus

$$e - \bar{e}(\rho) = c_\nu (T - \bar{T}(\rho)) \quad (80)$$

and

$$p(\rho, T) = \bar{p}(\rho) + \alpha c_\nu (T - \bar{T}(\rho)) . \quad (81)$$

Application of the second law of thermodynamics allows the determination of the variation of \bar{T} with ρ .

$$dS = \frac{de}{T} + p \frac{d\nu}{T} \quad (82)$$

$$= \frac{c_\nu}{T} dT + ((\partial e / \partial \nu)_T + p) \frac{d\nu}{T} \quad (83)$$

$$= \frac{c_\nu}{T} dT + (\partial p / \partial T)_\nu d\nu . \quad (84)$$

The identity $T(\partial p / \partial T)_\nu = (\partial e / \partial \nu)_T + p$ has been used in the final formula above. This identity follows from the consistency condition for dS to be an exact differential. Since $(\partial p / \partial T)_\nu = \alpha c_\nu$, Equation 84 is solvable on an isentrope. The solution of the differential equation for \bar{T} is

$$\bar{T} = T_\infty e^{-\alpha(\nu - \nu_\infty)} . \quad (85)$$

Integrating at constant volume to obtain the entropy, S , yields

$$S = \bar{S} + \int_{\bar{T}}^T \frac{c_\nu}{T} dT \quad (86)$$

$$= S_\infty + c_\nu \log(T / T_\infty) + \alpha c_\nu (\nu - \nu_\infty) . \quad (87)$$

A general relation for the sound speed is

$$c^2 = (\partial p / \partial \rho)_s = (\partial p / \partial \rho)_T + (\partial p / \partial T)_\rho (\partial T / \partial \rho)_s \quad (88)$$

$$= (\partial p / \partial \rho)_T + \frac{T}{\rho^2 c_\nu} (\partial p / \partial T)_\rho^2 . \quad (89)$$

Since $(\partial p / \partial T)_\rho = \alpha c_\nu$ and $(\partial p / \partial \rho)_T = d\bar{p} / d\rho - \alpha^2 \nu^2 c_\nu \bar{T}$,

$$c^2 = d\bar{p} / d\rho + \alpha^2 \nu^2 c_\nu (T - \bar{T}) . \quad (90)$$

The above equations should be sufficient to provide enough information to implement this equation of state in any code framework. Note that for the reference isentrope, $p(\rho_\infty, T_\infty) = 0$. Copper parameters used in this report are listed in Table 1 in cgs units. These were obtained from the copper table on page 532 of Appendix E of Kinslow [20]. Only the first three parameters are relevant to the isentropic flows presented in this report. The remaining parameters may be useful for comparison with codes utilizing the full equation of state listed above in this Appendix. Note that a linear isentrope may be obtained with $s = 0$ ($\gamma = -1$).

ρ_∞	8.94 gm/cc
c_∞	$3.94 \cdot 10^5$ cm/s
s	1.489 ($\gamma = 4.956$)
Γ	1.99
c_ν	$3.718 \cdot 10^6$ erg/(cm-deg K)
T_∞	293 deg K
e_∞	arbitrary
S_∞	arbitrary

Table 1: Reference parameter values for a copper equation of state.

B Evaluation of Continued Fractions

Continued fractions may be evaluated by the forward recursion algorithm for the numerator and denominator of the k -th approximate. That is, given the continued fraction

$$\beta_0 + \frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2} + \frac{\alpha_3}{\beta_3} + \cdots = \beta_0 + \frac{\alpha_1}{\beta_1 + \frac{\alpha_2}{\beta_2 + \cdots}}. \quad (91)$$

the numerator, A_k , and the denominator, B_k , of the k -th approximate, C_k , are given by

$$\begin{aligned} A_k &= \beta_k A_{k-1} + \alpha_k A_{k-2}, & B_k &= \beta_k B_{k-1} + \alpha_k B_{k-2}, & k &= 1, 2, 3, \dots \\ A_{-1} &= 1, & A_0 &= \beta_0, & B_{-1} &= 0, & B_0 &= 1 \end{aligned} \quad (92)$$

with

$$C_k = A_k / B_k. \quad (93)$$

If the continued fraction is convergent, $\lim_{k \rightarrow \infty} C_k$ exists and is defined as the value of the continued fraction. For fixed values of $\alpha_k = \tilde{\alpha}$ and $\beta_k = \tilde{\beta}$ is

$$\sigma^2 - \tilde{\beta}\sigma - \tilde{\alpha} = 0, \quad (94)$$

is the characteristic equation for the difference equation given in 92. The roots of this equation give the fundamental solutions σ_1^k and σ_2^k of the constant coefficient difference equation for A_k and B_k . A continued fraction is termed limit periodic if α_k and β_k approach constants $\tilde{\alpha}$ and $\tilde{\beta}$, respectively, as $k \rightarrow \infty$. The limit characteristic equation is given by Equation 94.

An acceleration technique for limit periodic continued fractions can be given by defining

$$C_k(w) = \frac{A_k - wA_{k-1}}{B_k - wB_{k-1}}. \quad (95)$$

If $w = \sigma_2$ and σ_2 is the subdominant root of the limit auxiliary equation ($|\sigma_2| < |\sigma_1|$), then $C_k(w)$ may converge faster than C_k . A thorough analysis of this approach is given by Thron and Waadeland [30]. In the present application the acceleration is not particularly fast due to the slow asymptotic approach to the limiting coefficients in the difference equation, i.e. $\alpha_k = \tilde{\alpha} + O(1/k)$, $\beta_k = \tilde{\beta} + O(1/k)$. As a result this technique was not utilized, but is included in this report for the sake of completeness as the FORTRAN code of Appendix E includes this option.

C Upper and Lower Bounds

The convergence theory for Chaplygin series is described by Sedov [29] and is based on analysis of the first order non-linear Ricatti equation derived from the linear second order differential equation for ψ_n . Let $Q = (2\tau/n)\psi'_n(\tau)/\psi_n(\tau)$ then the Ricatti equation for Q is

$$H(Q) = \frac{dQ}{d\tau} + \frac{M^2}{2\tau}Q + \frac{n}{2\tau}\{Q^2 - (1 - M^2)\} = 0 \quad (96)$$

with $Q(0) = 1$. To find upper and lower bounds for Q , concepts from the theory of differential inequalities may be applied. Suppose a comparison function $\bar{Q}(\tau)$ with $\bar{Q}(0) = 1$ can be found such that $H(\bar{Q}) \geq 0$ for $\tau \in [0, \tau_{cr}]$, then it is claimed that $Q \leq \bar{Q}$ for $\tau \in [0, \tau_{cr}]$. The proof follows by writing the differential equation for $Q - \bar{Q}$:

$$\frac{d}{d\tau}(Q - \bar{Q}) = -\frac{d\bar{Q}}{d\tau} - \frac{M^2}{2\tau}Q - \frac{n}{2\tau}\{Q^2 - (1 - M^2)\} \quad (97)$$

or, by expanding about \bar{Q} ,

$$\frac{d}{d\tau}(Q - \bar{Q}) = -H(\bar{Q}) - \left(\frac{M^2}{2\tau} + \frac{n}{\tau}\right)(Q - \bar{Q}) - \frac{n}{4\tau}(Q - \bar{Q})^2 \quad (98)$$

Thus, at any point $\tau \in [0, \tau_{cr}]$ such that $Q = \bar{Q}$, $d(Q - \bar{Q})/d\tau = -H(\bar{Q}) \leq 0$. It is thus impossible for $Q - \bar{Q}$ to be greater than zero. Similarly, for $H(\bar{Q}) \leq 0$, then $Q - \bar{Q} \geq 0$. In practice one finds a reasonable approximation, \bar{Q} , to the solution and evaluates $H(\bar{Q})$. If $H(\bar{Q}) \geq 0$, then \bar{Q} is an upper bound. If $H(\bar{Q}) \leq 0$, then \bar{Q} is a lower bound.

For $n \gg 1$, the equation for Q suggests the approximation

$$\bar{Q} = \sqrt{1 - M^2}. \quad (99)$$

One finds that

$$H(\bar{Q}) = \left(\tau \frac{d(\log \rho)}{d\tau} - \frac{M^4}{2\tau}\right) / \sqrt{1 - M^2} \leq 0 \quad (100)$$

provided $d\rho/d\tau \leq 0$. This assumption is physically correct since the density should decrease with increasing flow speed and lower pressure. Thus $\bar{Q} = \sqrt{1 - M^2}$ is a lower bound and leads directly by integration to Equation 42, the result giving the convergence proof for Chaplygin series.

We also found it useful to compute an upper bound solution of the form

$$\bar{Q} = \rho \sqrt{K + \epsilon^\delta \alpha(\tau)} \quad (101)$$

where $K = (1 - M^2)/\rho^2$ and $\epsilon = 1/n$. Substitution yields

$$H(\bar{Q}) = \frac{\rho}{2} \frac{dK/d\tau + \epsilon^\delta d\alpha/d\tau}{\sqrt{K + \epsilon^\delta \alpha(\tau)}} + \frac{\epsilon^{\delta-1} \rho^2 \alpha}{2\tau}. \quad (102)$$

Sedov (1965) chose $\alpha(\tau)$ to be $C\tau$ where C is a positive constant. The value of $\delta = 2/3$ was chosen in order to provide estimates on the sign of $H(\bar{Q})$ which were independent of n .

Assuming for the moment that $d\alpha/d\tau$ is bounded as ϵ goes to zero yields the dominant balance equation for α

$$\frac{\rho}{2} \frac{dK/d\tau}{\sqrt{K + \epsilon^\delta \alpha(\tau)}} + \frac{\epsilon^{\delta-1} \rho^2 \alpha}{2\tau} = 0 \quad (103)$$

or

$$\alpha^3 + \frac{K}{\epsilon^\delta} \alpha^2 - \epsilon^{2-3\delta} \left(-\frac{\tau}{\rho} \frac{dK}{d\tau} \right)^2 = 0. \quad (104)$$

The single non-negative real root of the above cubic is of interest. In order that α be independent of ϵ for $\tau = \tau_{cr}$, one specifies $\delta = 2/3$. Values of $\alpha(\tau)$ are easily found by Newton iteration, but may also be given explicitly:

$$\alpha = \frac{A^{1/3}}{2(a/3)^{1/2} \cos(\cos^{-1}(z)/3)} \quad \text{for } z \leq 1 \quad (105)$$

or

$$\alpha = \frac{A^{1/3}}{2(a/3)^{1/2} \cosh(\log(z + \sqrt{z^2 - 1})/3)} \quad \text{for } z \geq 1 \quad (106)$$

where $z = (3/a)^{3/2}/2$, $a = K\epsilon^{-2/3}A^{-1/3}$ and $A = (-\tau/\rho)dK/d\tau)^2$. Figure 31 shows values of $\alpha(\tau)$ for several values of n . The value of the functional H is then

$$H(\bar{Q}) = \frac{\rho}{2} \frac{\epsilon^{2/3} d\alpha/d\tau}{\sqrt{K + \epsilon^{2/3} \alpha(\tau)}} \geq 0 \quad (107)$$

since we show below that $d\alpha/d\tau \geq 0$. Thus \bar{Q} is an upper bound estimate. The behavior of $d\alpha/d\tau$ may be investigated by differentiating the cubic equation for α .

$$\frac{d\alpha}{d\tau} = \frac{\epsilon^{2/3} dA/d\tau - dK/d\tau \alpha^2}{3\epsilon^{2/3} \alpha^2 + 2K\alpha} \geq 0 \quad (108)$$

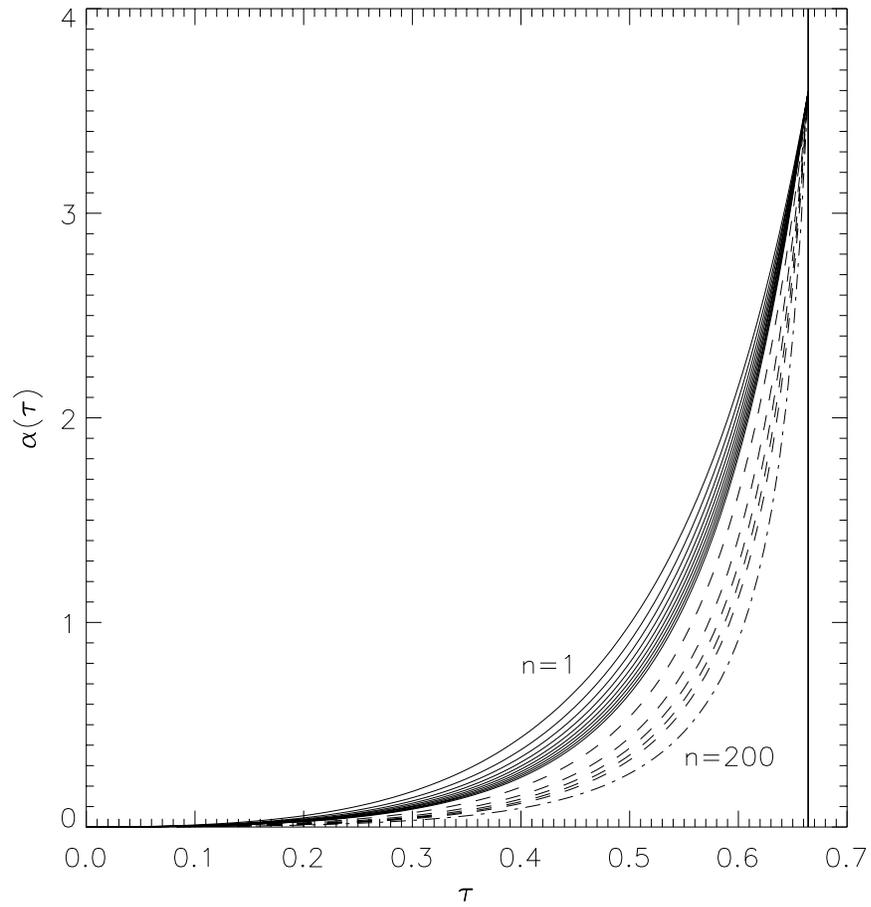


Figure 31: The function $\alpha(\tau)$ for $M = 1$ for the standard Cu isentrope. Solid - $n = 1$ to 10 by 1 ; dashed - $n = 20$ to 100 by 20 ; dash-dot - $n = 200$.

since it is assumed that $dK/d\tau \leq 0$. This is easy to show for the Murnaghan relations assumed in this report. For fixed $\tau < \tau_{cr}$ ($K \neq 0$), then it is seen that $H(\bar{Q}) = O(\epsilon^{2/3})$ as $\epsilon \rightarrow 0$. For fixed $\tau = \tau_{cr}$ ($K = 0$), then $H(\bar{Q}) = O(\epsilon^{-1/3})$ as $\epsilon \rightarrow 0$. This result indicates a basic non-uniformity inherent in the approximation to the differential equation. However, it is still interesting to inquire as to the magnitude of the error between Q and \bar{Q} . A careful comparison of the difference between the exact value of Q computed according to the methods outlined in the text and the upper bound approximation, \bar{Q} , has yielded convincing numerical evidence that $\bar{Q}(\tau_{cr}) - Q(\tau_{cr}) = O(\epsilon^{1/2})$ as $\epsilon \rightarrow 0$. This suggests that the function \bar{Q} is a leading order asymptotic approximation to Q as well as an upper bound since it would follow that $(\bar{Q} - Q)/\bar{Q} = O(\epsilon^{1/6})$.

D Several Sequence Transformations

Although the θ , Levin-u and Levin-t algorithms were not successful in accelerating the complex sequences given by the partial sums of z_q and z_n for all needed parameter values, a short summary of these algorithms is given here as the coding is included in the program listing.

The θ -algorithm is given by

$$\theta_{-1}^{(m)} = 0, \quad \theta_0^{(m)} = s_m, \quad m \geq 0 \quad (109)$$

$$\theta_{2k+1}^{(m)} = \theta_{2k-1}^{(m+1)} + (\theta_{2k}^{(m+1)} - \theta_{2k}^{(m)})^{-1}, \quad k \geq 0, \quad (110)$$

$$\theta_{2k+2}^{(m)} = \frac{\theta_{2k}^{(m+2)} \Delta \theta_{2k+1}^{(m+1)} - \theta_{2k}^{(m+1)} \Delta \theta_{2k+1}^{(m)}}{\Delta^2 \theta_{2k+1}^{(m)}}, \quad k \geq 0. \quad (111)$$

Only the θ_{2k}^m terms are used as estimates for the sum.

The Levin transforms are given by

$$t_k(s_m) = t_k^{(m)} = \frac{\sum_{j=0}^k (s_{m+j}/a_{m+j+1})(m+j+1)^{k-1}(-1)^j \binom{k}{j}}{\sum_{j=0}^k (1/a_{m+j+1})(m+j+1)^{k-1}(-1)^j \binom{k}{j}} \quad (112)$$

and

$$u_k(s_m) = u_k^{(m)} = \frac{\sum_{j=0}^k (s_{m+j}/a_{m+j+1})(m+j+1)^{k-2}(-1)^j \binom{k}{j}}{\sum_{j=0}^k (1/a_{m+j+1})(m+j+1)^{k-2}(-1)^j \binom{k}{j}} \quad (113)$$

where a_j are terms of the series leading to the partial sums $s_m = \sum_{j=0}^m a_j$ for $m \geq 0$. The u-transform and the t-transform may be evaluated using a similar Lozenge type diagram just as in the case of the ϵ and θ -algorithms. Define for all $m \geq 0$

$$Q_0^{(m)} = \frac{s_m}{(m+1)^p a_{m+1}} \quad (114)$$

$$Q_k^{(m)} = Q_{k-1}^{(m+1)} - \left(\frac{m+1}{m+k+1} \right) \left(\frac{m+k}{m+k+1} \right)^{k-1} Q_{k-1}^{(m)}, \quad k \geq 1 \quad (115)$$

and

$$\tilde{Q}_0^{(m)} = \frac{1}{(m+1)^p a_{m+1}} \quad (116)$$

$$\tilde{Q}_k^{(m)} = \tilde{Q}_{k-1}^{(m+1)} - \left(\frac{m+1}{m+k+1} \right) \left(\frac{m+k}{m+k+1} \right)^{k-1} \tilde{Q}_{k-1}^{(m)}, \quad k \geq 1 \quad (117)$$

then

$$t_k^{(m)} = Q_k^{(m)} / \tilde{Q}_k^{(m)} \quad \text{with } p = 1 \quad (118)$$

$$u_k^{(m)} = Q_k^{(m)} / \tilde{Q}_k^{(m)} \quad \text{with } p = 2. \quad (119)$$

The above evaluation algorithm is derived by Fessler, Ford and Smith (1983) for the u-algorithm and easily generalizes for the t-transform.

E Verification testing using CJETB

When using a general purpose numerical code to compute the wall jet solutions, it is possible to verify some aspects of the solutions by checking simple consistency conditions. For example, in the case of a normal incidence free surface wall jet of subsonic inflow velocity ($\beta = 90^\circ$), one can easily check a necessary condition for the computed steady state flow to be correct. Since the steady flow is isentropic and thus reversible, there must be a symmetry of all state variables about a 45° line in the first quadrant. Thus, for this case, *any* reasonable equation of state can be used for testing. One must be careful to check all variables as it has been found, in one instance, that the pressure profile computed was highly symmetric, while the internal energy and temperature profiles were quite asymmetric indicating non-physical dissipation in a subsonic isentropic flow.

Setting up a useful verification problem using a transient flow shock capturing code may require some ingenuity since the solution presented here is steady state. Since the flow has a stagnation point, it will take some time for any initial transient response to be advected out of the problem. Another possibility is to implement an approximate initial condition based on the incompressible solution. The steady state defined by this exact solution could also be used as a test of a mesh to mesh interpolation capability and thus would also be useful for setting up a precise initial steady state flow distribution to avoid the initial transient problem.

A copy of the relevant code, CJETB, and related subroutines used to evaluate the wall jet solution described in the text is given below. The code

writes out some data in the EXODUS I finite element database format as well as x-y pair data for plotting [21]. It is expected that any user interested in running the CJETB code will have access to reasonable postprocessing procedures for this type of data and may modify the code with little effort to be compatible with any particular finite element database format.

```

1 C *****
2 C
3 C           CJETB (Compressible JET (Blot compatible output) )
4 C
5 C           Two Dimensional Output in EXODUS I Finite Element Format
6 C           BLOT may be used to display this file
7 C
8 C           1D output in ASCII tables
9 C
10 C *****
11 C
12 C Author:
13 C           Allen C. Robinson
14 C           Sandia National Laboratories
15 C           MS 0819
16 C           Albuquerque, NM 87122-0819
17 C
18 C           acrobin@sandia.gov
19 C
20 C Background: This program evaluates the solution to the
21 C               problem of a plane compressible free surface jet
22 C               impinging at any angle to a fixed wall. The incompressible
23 C               complex potential for this problem is known in closed form
24 C               and leads immediately to the solution for a subsonic
25 C               compressible jet by Chaplygin's method.
26 C               The series representation for this solution
27 C               converges very slowly for points on and near the free
28 C               surface. This difficulty is overcome using non-linear
29 C               convergence accelerators. We assume a Murnaghan equation
30 C               of state leading to hypergeometric functions as solutions
31 C               to the reduced Chaplygin equation. The functions F are
32 C               evaluated using continued fraction representations
33 C               for F'/F. Log F is then obtained
34 C               by numerical integration. The physical plane z may be computed
35 C               by two schemes. The first computes dz/dq by summing series
36 C               and then integrating numerically with respect to q.
37 C               Alternatively z may be computed
38 C               in terms of an analytical integration of dz/d(theta).
39 C               Velocity, pressure, internal energy, density, Mach number,
40 C               and several strain rate measures are computed as a function
41 C               of the physical plane position z.
42 C
43 C Disclaimer:
44 C
45 C           *****
46 C           *   ISSUED BY SANDIA   *
47 C           * NATIONAL LABORATORIES *
48 C           *   A PRIME CONTRACTOR *
49 C           *****   TO THE   *

```

```

50 C          * UNITED STATES *
51 C          * DEPARTMENT *
52 C          * OF *
53 C          * ENERGY *
54 C ***** --NOTICE--- *****
55 C * THE UNITED STATES GOVERNMENT RETAINS, IN THIS SOFTWARE, *
56 C * A PAID-UP, NONEXCLUSIVE, IRREVOCABLE WORLDWIDE LICENSE *
57 C * TO REPRODUCE, PREPARE DERIVATIVE WORKS, PERFORM PUBLICLY *
58 C * AND DISPLAY PUBLICLY BY OR FOR THE GOVERNMENT, INCLUDING *
59 C * THE RIGHT TO DISTRIBUTE TO OTHER GOVERNMENT CONTRACTORS. *
60 C * NEITHER THE UNITED STATES, THE UNITED STATES DEPARTMENT *
61 C * OF ENERGY, NOR ANY OF THEIR EMPLOYEES, MAKES ANY WARRANTY, *
62 C * EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR *
63 C * ***** RESPONSIBILITY ***** *
64 C * * * FOR THE * * *
65 C * * * ACCURACY, * * *
66 C * * * COMPLETENESS * * *
67 C * * * OR USEFULNESS * * *
68 C * * * OF ANY * * *
69 C * * * INFORMATION, * * *
70 C * ***** * APPARATUS, * ***** *
71 C * * * PRODUCT * * *
72 C * * * OR PROCESS * * *
73 C * * * DISCLOSED, * * *
74 C * * * OR REPRESENTS * * *
75 C * * * ** THAT ITS ** * *
76 C * * * ** USE WOULD NOT ** * *
77 C ***** ** INFRINGE ** *****
78 C ** PRIVATELY **
79 C ** OWNED **
80 C ** RIGHTS. **
81 C ** **
82 C ** **
83 C *****
84 C
85 C
86 C
87 PROGRAM CJETB
88 PARAMETER(NQMAX=100, NUMAX=200, NTMAX=100, NGMAX=2, NVMAX=24, KMAX=201)
89 PARAMETER(NQNTMX=NQMAX*NTMAX)
90 COMPLEX EITHTA, ZDQ, ZN, ZSEST, ZTMP, ZS (NUMAX, KMAX)
91 EXTERNAL ZDQ, ZN
92 CHARACTER*8 NAMEGV (NGMAX), NAMENV (NVMAX), NAMECO (2), NAMELB (1)
93 CHARACTER*8 CNAME1, CNAME2
94
95 character*25 runid
96 character*8 cdate
97 character*10 ctime
98 character*5 czone
99 integer dtval(8)
100
101 CHARACTER*10 SUMFLG
102 CHARACTER*80 CHAR
103 COMMON /FLOCOM/ IQ, NQPNTS, EITHTA, PI, SNTH (NUMAX), CNTH (NUMAX)
104 COMMON /GASCOM/ GAMMA, GM1D2, QMAX2, QCR2, TAUCR, Q1, TAU1, PSI2T1,
105 1 Q (NQMAX), TAU (NQMAX), RHO (NQMAX),
106 2 ENERGY (NQMAX), PRES (NQMAX), XMACH2 (NQMAX),

```

```

107      3          FPDF(NUMAX,NQMAX),XLNF(NUMAX,NQMAX),
108      4          F(NUMAX,NQMAX),FP(NUMAX,NQMAX),
109      5          PSI(NUMAX,NQMAX),PSIP(NUMAX,NQMAX),
110      6          PSIRAT(NUMAX,NQMAX),NCHPSM(NUMAX,NQMAX)
111      DIMENSION XDQ(NTMAX,NQMAX),YDQ(NTMAX,NQMAX)
112      DIMENSION XT(NTMAX,NQMAX),YT(NTMAX,NQMAX)
113      DIMENSION X(NTMAX,NQMAX),Y(NTMAX,NQMAX)
114      DIMENSION XO(NTMAX,NQMAX),YO(NTMAX,NQMAX),DIS(NTMAX,NQMAX)
115      DIMENSION U(NTMAX,NQMAX),V(NTMAX,NQMAX)
116      DIMENSION DIVU(NTMAX,NQMAX)
117      DIMENSION UX(NTMAX,NQMAX),UYORVX(NTMAX,NQMAX),VY(NTMAX,NQMAX)
118      DIMENSION XMAGDU(NTMAX,NQMAX),DEVU(NTMAX,NQMAX)
119      DIMENSION TMPA(4),TMPB(4)
120      DIMENSION THETA(NTMAX),TX(3),TF(3),TDF(3)
121      DIMENSION NEST(NTMAX,NQMAX,2,4),KEST(NTMAX,NQMAX,2,4)
122      DIMENSION OUT(NQNTMX,2),ICONK(4,NQNTMX)
123      NAMELIST /EOS/ RHOINF,CINF,SINF,EINF
124      DATA RHOINF/8.94/,CINF/3.94E+5/,SINF/1.489/,EINF/0./
125      NAMELIST /JET/ XLINF,XMCHMN,XMCHMX,NMACH,BETDMN,BETDMX,NBETA
126      DATA XLINF/1./,XMCHMN/.9/,XMCHMX/.9/,NMACH/1/
127      DATA BETDMN/90./,BETDMX/90./,NBETA/1/
128      NAMELIST /TEKNIK/ NTPNTS,NQPNTS,NQSW,RELERR,SSLP,
129      1  RELER1,NUBIG,KBIG,ISUM,INTFLG,IUNIT
130      DATA NTPNTS/51/,NQPNTS/51/,NQSW/0/,RELERR/1.E-6/,SSLP/0./
131      DATA RELER1/1.E-4/,NUBIG/200/,KBIG/201/
132      DATA ISUM/2/,INTFLG/1/,IUNIT/1/
133      DATA CNAME1/'CJETB',CNAME2/'V1.0'/
134
135      PI = 4.*ATAN(1.)
136
137 C      get first argument in list
138      call getarg ( 1, runid)
139      do 10 idlen=1,len(RUNID)-4
140          if(runid(idlen:idlen).eq.' ') then
141              goto 11
142          endif
143      10 continue
144      11 continue
145
146 C Unit numbers: input = 5
147 C      output log =6
148 C      On axis data output in GRAPH format = 8
149 C      Chapylgin function output in GRAPH format =10
150 C      Full data output =11 in EXODUS format for use with BLOT
151 C      postprocessing graphics.
152 C Respective file suffixes are runid.inp,runid.out,runid.axs,runid.chp,runid.exo
153
154      NAB = 8
155      NCB = 10
156      NDB = 11
157
158 C To compute the tables of Ferguson and Lighthill for subsonic tau
159 C then input NQPNTS = 9, NQSW=1, SINF=.6 (GAMMA=1.4) and XMCHMN=SQRT(20./21.)'
160 C The value SINF = 0. is the Chaplygin gas.
161
162      runid(idlen:idlen+3)='.inp'
163      OPEN(UNIT=5,FILE=runid,STATUS='old',FORM='formatted')

```

```

164      READ(5,EOS)
165 C      RHOINF=freestreamline density
166 C      CINF=freestreamline soundspeed
167 C      SINP=value of s in Us-Up relation
168 C      EINF=freestreamline internal energy
169 C
170      READ(5,JET)
171 C      NAMELIST /JET/ XLINF,XMCHMN,XMCHMX,NMACH,BETDMN,BETDMX,NBETA
172 C      XLINF=width of incoming jet at infinity
173 C      XMCHMN=minimum value of Mach number (0<XMCHMN<1)
174 C      XMCHMX=maximum value of Mach number (0<XMCHMX<1)
175 C      NMACH=number of Mach numbers to compute
176 C      BETDMN=minimum value of BETA in degrees (0<BETDMN<180)
177 C      BETDMX=maximum value of BETA in degress (0<BETDMX<180)
178 C      NBETA=number of BETA values to compute
179      READ(5,TEKNIK)
180 C      NTPNTS=number of evaluation points in theta direction
181 C      NQPNTS=number of evaluation and trapezoidal rule integration
182 C      points in Q direction
183 C      NQSW=switch to change from constant dQ to constant dtau
184 C      RELERR=relative error check value for Chaplygin function evaluation
185 C      RELERR1=relative error check value for series summation
186 C      SSLP=slope of stretch mapping near singular points (0.LE.SSLP.LE.1.)
187 C      NUBIG=maximum number of terms in column summation
188 C      KBIG=maximum row value in Lozenge scheme for acceleration schemes.
189 C      ISUM=acceleration sum type = 1      2      3      4      5
190 C      for valid SUMFLG values =NONE, EPSILON, THETA, LEVINU, LEVINT
191 C      INTFLG=type of integration for basic physical plane
192 C      =1 Q integration
193 C      =2 theta integration
194 C      IUNIT =output scaling
195 C      =1 for dimensional units
196 C      =2 for stagnation point scaling
197 C      =3 for freestreamline scaling
198 C      See code interior for definitions
199      IF(NQSW.EQ.0) NQSW=NQPNTS/5
200      NQSW=MAX(1,MIN(NQSW,NQPNTS))
201      IF(NUBIG.GT.NUMAX) THEN
202          WRITE(6,*) 'NUBIG .GT. NUMAX'
203          STOP
204      ENDIF
205      IF(KBIG.GT.KMAX) THEN
206          WRITE(6,*) 'KBIG .GT. KMAX'
207          STOP
208      ENDIF
209      IF(ISUM.EQ.1) SUMFLG='NONE'
210      IF(ISUM.EQ.2) SUMFLG='EPSILON'
211      IF(ISUM.EQ.3) SUMFLG='THETA'
212      IF(ISUM.EQ.4) SUMFLG='LEVINU'
213      IF(ISUM.EQ.5) SUMFLG='LEVINT'
214      IF(ISUM.LT.1.OR.ISUM.GT.5) THEN
215          WRITE(6,*) 'BAD VALUE OF ISUM'
216          STOP
217      ENDIF
218      IF(INTFLG.NE.1.AND.INTFLG.NE.2) THEN
219          WRITE(6,*) 'BAD VALUE OF INTFLG'
220          STOP

```

```

221         ENDIF
222         IF (IUNIT.LT.1.AND.IUNIT.GT.3) THEN
223             WRITE(6,*) 'BAD VALUE OF IUNIT'
224             STOP
225         ENDIF
226     CLOSE(UNIT=5)
227
228     runid(idlen:idlen+3)='.out'
229
230     call date_and_time(cdate,ctime,czone,dtval)
231
232     WRITE(6,*) 'Begin code ',CNAME1,' ',CNAME2,' at ',
233     1    CTIME,' on ',CDATE,','
234
235     WRITE(6,EOS)
236     WRITE(6,JET)
237     WRITE(6,TEKNIK)
238
239     GAMMA=4.*SINF-1.
240     XKPINF=RHOINF*CINF**2/GAMMA
241
242     WRITE(6,*) 'XKPINF and GAMMA chosen to fit slope and curvature'
243     WRITE(6,*) 'XKPINF = ',XKPINF,' GAMMA = ',GAMMA
244     IF (GAMMA.GT.-1.AND.GAMMA.LE.1.) THEN
245         WRITE(6,*) 'THIS VALUE OF GAMMA NOT ALLOWED'
246         STOP
247     ENDIF
248
249 C compute various functions of GAMMA
250     GM1D2=(GAMMA-1.)/2.
251     QMAX2=1./GM1D2
252     IF (GAMMA.LE.-1.) THEN
253         QCR2 = 1.E+30
254     ELSE
255         QCR2 =2./(GAMMA+1.)
256     ENDIF
257     TAUCR=QCR2/QMAX2
258
259 C ***** Begin Mach Number Loop *****
260
261     DO 5000 IMVARY=1,NMACH
262
263         XMACHI = XMCHMN
264         IF (IMVARY.GT.1)
265     1    XMACHI=(XMCHMX-XMCHMN)*FLOAT(IMVARY-1)/FLOAT(NMACH-1)+XMCHMN
266         WRITE(6,*) 'XMACHI = ',XMACHI
267
268         XMCHI2=XMACHI**2
269         Q1=SQRT(XMCHI2/(1.+GM1D2*XMCHI2))
270         TAU1=Q1**2/QMAX2
271
272 C Compute stagnation point parameters in dimensional units
273         TMP = 1.+GM1D2*XMCHI2
274         RHO0= RHOINF*TMP**(1./(GAMMA-1.))
275         CO = CINF*SQRT(TMP)
276         XKPO = RHO0*CO**2/GAMMA
277         PO = XKPO-XKPINF

```

```

278      EO = (XKPO/RHOO-XKPINF/RHOINF)/(GAMMA-1.)
279      1  + XKPINF*(1./RHOO - 1./RHOINF) + EINF
280
281 C Output stagnation points values
282
283      WRITE(6,*) 'Stagnation point density =',RHOO
284      WRITE(6,*) 'Stagnation point sound speed =',CO
285      WRITE(6,*) 'Stagnation point pressure =',PO
286      WRITE(6,*) 'Stagnation point internal energy =',EO
287
288 C Basic variables are in stagnation point units.
289 C Pressure and energy are in input units.
290
291      DO 500 J = 1,NQPNTS
292
293          Q(J) = (J-1)*Q1/FLOAT(NQPNTS-1)
294          TAU(J)=Q(J)**2/QMAX2
295          IF(J.GT.NQSW) THEN
296              TAU(J) =
297      1      (J-NQSW)*(TAU1-TAU(NQSW))/FLOAT(NQPNTS-NQSW)+TAU(NQSW)
298          Q(J)=SQRT(QMAX2*TAU(J))
299          ENDIF
300
301          TMP=1.-TAU(J)
302          RHO(J)=TMP**(1./(GAMMA-1.))
303          PRES(J)=(XKPO*RHO(J)**GAMMA-XKPINF)
304          ENERGY(J)=(XKPO*TMP/RHOO-XKPINF/RHOINF)/(GAMMA-1.)
305      1      +XKPINF*(1./(RHOO*RHO(J))-1./RHOINF) +EINF
306          XMACH2(J)=Q(J)**2/TMP
307 500    CONTINUE
308 C      Set roundoff to zero
309          PRES(NQPNTS)=0.
310          ENERGY(NQPNTS)=EINF
311
312 C Compute Chaplygin functions for this value of Mach number
313
314          CALL CHPLGN(NQPNTS,RELERR)
315          WRITE(6,*) 'CHAPLYGIN FUNCTIONS COMPUTED TO ORDER ',NUMAX
316
317          runid(idlen:idlen+3)=' .chp'
318          OPEN(UNIT=NCB,FILE=runid,STATUS='unknown',FORM='formatted')
319          WRITE(NCB,557) GAMMA,TAU1,TAUCR
320 557    FORMAT(' !CHAPLYGIN FUNCTIONS FOR GAMMA,TAU1,TAUCR=',3E15.6)
321          DO 555 NU=1,NUMAX
322              WRITE(NCB,558) NU
323 558    FORMAT(' !CHAPLYGIN FUNCTIONS OF ORDER =',I5,/,
324      1    ' !      TAU      $      FPDF      $      XLNF      $',
325      2    '      F      $      FP      $      PSI/PSI1      $',
326      3    '      PSIP/PSI1      $(2T/N)PSIP/PSI$      NCHPSM$')
327          DO 550 J=1,NQPNTS
328              WRITE(NCB,560)TAU(J),FPDF(NU,J),XLNF(NU,J),F(NU,J),FP(NU,J),
329      1      PSI(NU,J),PSIP(NU,J),PSIRAT(NU,J),NCHPSM(NU,J)
330 560    FORMAT(1X,8E15.6,I10)
331 550    CONTINUE
332          WRITE(NCB,('' $ $ $ $ $ $ $ $ $ $'))
333 555    CONTINUE
334          CLOSE(UNIT=NCB)

```

```

335     WRITE(6,*) 'CHAPLYGIN FUNCTIONS WRITTEN TO CHPFNC'
336
337
338 C ***** BETA variation *****
339
340     DO 5000 IBVARY=1,NBETA
341
342         BETAD=BETDMN
343         IF (IBVARY.GT.1)
344     1   BETAD=(BETDMX-BETDMN)*FLOAT(IBVARY-1)/FLOAT(NBETA-1)+BETDMN
345         WRITE(6,*) 'BETAD = ',BETAD
346
347         BETA=BETAD*PI/180.
348
349         TMPB(1)=1.
350         TMPB(2)=1.
351         TMPB(3)=- (1.-COS(BETA))
352         TMPB(4)=- (1.+COS(BETA))
353
354     DO 900 J=1,NQPNTS
355     DO 900 I=1,NTPNTS
356         XDQ(I,J)=0.
357         YDQ(I,J)=0.
358         XT(I,J)=0.
359         YT(I,J)=0.
360 900 CONTINUE
361
362     TMPA(1)=-BETA+PI
363     TMPA(2)=BETA-PI
364     TMPA(3)=0.
365     TMPA(4)=-PI
366
367 C     use equal number of theta points on each side of singularity
368     IBETA=(NTPNTS+1)/2
369
370 C Stretch coordinates to get better coverage.
371 C Stretch slope 0.LE.SSLP.LE.1.
372
373     TX(1) = 0.
374     TX(2) = PI-BETA
375     TX(3) = PI
376     TF(1) = 0.
377     TF(2) = PI-BETA
378     TF(3) = PI
379     TDF(1)= SSLP
380     TDF(2)= SSLP
381     TDF(3)= SSLP
382
383     DO 950 I = 1,NTPNTS
384         IF (I.LE.IBETA) THEN
385             TMP=(TF(2)-TF(1))*(I-1)/FLOAT(IBETA-1)+TF(1)
386             CALL HERMIT(TX,TF,TDF,2,2,TMP,TMP1,TMP2)
387         ELSE
388             TMP=(TF(3)-TF(2))*(I-IBETA)/FLOAT(NTPNTS-IBETA)+TF(2)
389             CALL HERMIT(TX(2),TF(2),TDF(2),2,2,TMP,TMP1,TMP2)
390         ENDIF
391         THETA(I)=-TMP1

```

```

392 950 CONTINUE
393
394 C Summation loops
395 C L is the loop for the four different infinite sums
396 C I loops through the various values of theta
397 C IQ is the flow speed loop
398
399 DO 1000 L=1,4
400 DO 1000 I = 1,NTPNTS
401 EITHTA=CMPLX(COS(THETA(I)),SIN(THETA(I)))
402 TMP1=THETA(I)+TMPA(L)
403 DO 1100 K=2,NUBIG
404 CNTH(K)=COS(K*TMP1)
405 SNTH(K)=SIN(K*TMP1)
406 1100 CONTINUE
407 DO 1000 IQ=1,NQPNTS
408 IF (IQ.EQ.NQPNTS.AND.(I.EQ.1.OR.I.EQ.NTPNTS.OR.I.EQ.IBETA)) THEN
409 NEST(I,IQ,1,L)=0
410 KEST(I,IQ,1,L)=0
411 NEST(I,IQ,2,L)=0
412 KEST(I,IQ,2,L)=0
413 GOTO1000
414 ENDIF
415 CALL CSUM(ZDQ,ZS,NUMAX,ZSEST,RELER1,
416 1 NEST(I,IQ,1,L),KEST(I,IQ,1,L),NUBIG,KBIG,SUMFLG,IERR)
417 IF(IERR.EQ.1) WRITE(6,*) 'ZDQ SUM FAILURE AT I,IQ,L=',I,IQ,L
418 ZSEST=TMPB(L)*ZSEST
419 XDQ(I,IQ)=REAL(ZSEST)+XDQ(I,IQ)
420 YDQ(I,IQ)=AIMAG(ZSEST)+YDQ(I,IQ)
421 C sum integration wrt theta
422 CALL CSUM(ZN,ZS,NUMAX,ZSEST,RELER1,
423 1 NEST(I,IQ,2,L),KEST(I,IQ,2,L),NUBIG,KBIG,SUMFLG,IERR)
424 IF(IERR.EQ.1) WRITE(6,*) 'ZN SUM FAILURE AT I,IQ,L=',I,IQ,L
425 ZSEST=TMPB(L)*ZSEST
426 XT(I,IQ)=REAL(ZSEST)+XT(I,IQ)
427 YT(I,IQ)=AIMAG(ZSEST)+YT(I,IQ)
428 1000 CONTINUE
429
430 C sum using trapezoidal rule
431 DO 2000 I=1,NTPNTS
432 X(I,1)=0.
433 Y(I,1)=0.
434 DIS(I,1)=0.
435 DO 2000 J=2,NQPNTS
436 X(I,J)=X(I,J-1)+.5*(Q(J)-Q(J-1))*(XDQ(I,J)+XDQ(I,J-1))
437 Y(I,J)=Y(I,J-1)+.5*(Q(J)-Q(J-1))*(YDQ(I,J)+YDQ(I,J-1))
438 DIS(I,J)=(X(I,J)-XT(I,J))**2+(Y(I,J)-YT(I,J))**2
439 IF(DIS(I,J).NE.0.) DIS(I,J)=SQRT(DIS(I,J))
440 2000 CONTINUE
441
442 IF(INTFLG.EQ.2) THEN
443 C Use theta integration grid for plots
444 DO 2100 J=1,NQPNTS
445 DO 2100 I=1,NTPNTS
446 X(I,J)=XT(I,J)
447 Y(I,J)=YT(I,J)
448 2100 CONTINUE

```

```

449         ENDIF
450
451 C Save first physical plane as the "undistorted mesh". Different values
452 C of the mach number and beta come out at different "times" in the EXODUS
453 C format.
454
455         IF (IMVARY.EQ.1.AND.IBVARY.EQ.1) THEN
456             DO 2200 J=1,NQPNTS
457                 DO 2200 I=1,NTPNTS
458                     XO(I,J)=X(I,J)
459                     YO(I,J)=Y(I,J)
460 2200             CONTINUE
461         ENDIF
462
463 C evaluate velocity field and gradients
464
465         DO 3100 J=1,NQPNTS
466         DO 3100 I=1,NTPNTS
467             IF (J.EQ.NQPNTS.AND.(I.EQ.1.OR.I.EQ.NTPNTS.OR.I.EQ.IBETA))
468 1          GOTO 3100
469             TMP1=COS(THETA(I))
470             TMP2=SIN(THETA(I))
471             U(I,J)=Q(J)*TMP1
472             V(I,J)=Q(J)*TMP2
473             EITHTA=CMPLX(TMP1,TMP2)
474             ZTMP = CMPLX(XDQ(I,J),YDQ(I,J))/EITHTA
475             AQ=REAL(ZTMP)
476             BQ=AIMAG(ZTMP)
477             ZTMP=EITHTA**2
478             C2T=REAL(ZTMP)
479             S2T=AIMAG(ZTMP)
480             DIVIDE=AQ**2+(1.-XMACH2(J))*BQ**2
481             A1=.5*XMACH2(J)*AQ
482             A2=.5*(2.-XMACH2(J))*AQ
483             A3=(1.-XMACH2(J))*BQ
484             UX(I,J)=(A1+C2T*A2-S2T*A3)/DIVIDE
485             UYORVX(I,J)=(C2T*A3+S2T*A2)/DIVIDE
486             VY(I,J)=(A1-C2T*A2+S2T*A3)/DIVIDE
487             DIVU(I,J)=UX(I,J)+VY(I,J)
488 C         to make plots come out nice set divu = 0 on free surface
489 C         actual values are zero to within numerical and roundoff errors
490             IF (J.EQ.NQPNTS) DIVU(I,J)=0.
491             XMAGDU(I,J)=SQRT(UX(I,J)**2+VY(I,J)**2+2.*UYORVX(I,J)**2)
492             DEVU(I,J)=SQRT(XMAGDU(I,J)**2-DIVU(I,J)**2/3.)
493 3100         CONTINUE
494
495 3200     CONTINUE
496
497 C Output units flag
498 C dimensional units = 1
499 C stagnation point scaling = 2
500 C and free-streamline scaling = 3
501
502 C dimensional units
503         IF (IUNIT.EQ.1) THEN
504             DUNITS=RHOO
505             XUNITS=XLINF

```

```

506         VUNITS=C0
507         GUNITS=C0/XLINF
508         PUNITS=1.
509         EUNITS=1.
510     ENDIF
511 C stagnation points scaling: scale density by RHO0
512 C                             scale velocities by C0
513 C                             scale lengths by XLINF
514 C                             scale pressure by P0
515 C                             scale energy by E0
516     IF(IUNIT.EQ.2) THEN
517         DUNITS=1.
518         XUNITS=1.
519         VUNITS=1.
520         GUNITS=1.
521         PUNITS=1./P0
522         EUNITS=1./E0
523     ENDIF
524 C freestreamline scaling: scale density by RHOINF
525 C                             scale velocities by freestreamline velocity, vinf
526 C                             scale lengths by XLINF
527 C                             scale pressure by .5*RHOINF*vinf**2
528 C                             scale energy by .5*vinf**2
529     IF(IUNIT.EQ.3) THEN
530         DUNITS=RHO0/RHOINF
531         XUNITS=1.
532         VUNITS=C0/(C0*Q1)
533         GUNITS=VUNITS/XUNITS
534         PUNITS=1./(.5*RHOINF*(C0*Q1)**2)
535         EUNITS=1./(.5*(C0*Q1)**2)
536     ENDIF
537
538     IF(IVARY.EQ.1.AND.IMVARY.EQ.1) THEN
539         CHAR=SUMFLG
540         CHAR(8:10)=' ZQ'
541         IF(INTFLG.EQ.2) CHAR(8:10)=' ZN'
542         CHAR(11:17)=' DMNSNL'
543         IF(IUNIT.EQ.2) CHAR(11:17)=' STGNTN'
544         IF(IUNIT.EQ.3) CHAR(11:17)=' FRSTRM'
545         WRITE(CHAR(18:24),'('' NQ='',I3)') NQPNTS
546         WRITE(CHAR(25:31),'('' NT='',I3)') NTPNTS
547         WRITE(CHAR(32:38),'('' NS='',I3)') NUBIG
548         WRITE(CHAR(39:45),'('' KS='',I3)') KBIG
549
550 C         Use EXODUS finite element output format
551 C         Meaningless values are written in a number of locations
552 C         since these are unnecessary
553
554         runid(idlen:idlen+3)='.exo'
555         OPEN (UNIT=NDB,FILE=runid,STATUS='unknown',
556     1         FORM='unformatted')
557         WRITE (NDB) CHAR
558         NUMNOD=NTPNTS*NQPNTS-3
559         NDIM=2
560         NUMEL=(NTPNTS-1)*(NQPNTS-1)-4
561         NELBLK=1
562         NUMNPS=1

```

```

563         LNPSNL=1
564         NUMESS=1
565         LESSEL=1
566         LESSNL=1
567         WRITE (NDB) NUMNOD, NDIM, NUMEL, NELBLK,
568 1         NUMNPS, LNPSNL, NUMESS, LESSEL, LESSNL, 1
569         DO 4100 K=1,2
570         ITMP=0
571         DO 4100 J=1, NQPNTS
572         DO 4100 I=1, NTPNTS
573             IF (J.EQ. NQPNTS .AND. (I.EQ. 1 .OR. I.EQ. NTPNTS .OR. I.EQ. IBETA))
574 1             GOTO 4100
575             ITMP=ITMP+1
576             OUT(ITMP, K)=(2-K)*X(I, J)+(K-1)*Y(I, J)
577 4100         CONTINUE
578         WRITE(NDB) ((OUT(I, J), I=1, NUMNOD), J=1, NDIM)
579         DO 4200 I=1, NUMEL
580 4200         ICONK(1, I)=I
581         WRITE(NDB)((ICONK(J, I), J=1, 1), I=1, NUMEL)
582         NATRIB=1
583         WRITE (NDB) 1, NUMEL, 4, NATRIB
584         ITMP=0
585         DO 4300 J=1, NQPNTS-1
586         DO 4300 I=1, NTPNTS-1
587             ITMP2=I+J*NTPNTS
588             IF (J.EQ. NQPNTS-1) THEN
589                 IF (I.EQ. 1) GOTO 4300
590                 IF (I.EQ. IBETA-1 .OR. I.EQ. IBETA) GOTO 4300
591                 IF (I.EQ. NTPNTS-1) GOTO 4300
592                 ITMP2=ITMP2-1
593                 IF (I.GT. IBETA) ITMP2=ITMP2-1
594             ENDIF
595             ITMP1=I+(J-1)*NTPNTS
596             ITMP4=ITMP1+1
597             ITMP3=ITMP2+1
598             ITMP=ITMP+1
599             ICONK(1, ITMP)=ITMP1
600             ICONK(2, ITMP)=ITMP2
601             ICONK(3, ITMP)=ITMP3
602             ICONK(4, ITMP)=ITMP4
603 4300         CONTINUE
604         WRITE(NDB) ((ICONK(J, I), J=1, 4), I=1, NUMEL)
605         WRITE(NDB) ((OUT(I, J), J=1, NATRIB), I=1, NUMEL)
606         WRITE(NDB) (ICONK(J, 1), J=1, NUMNPS)
607         WRITE(NDB) (ICONK(J, 1), J=1, NUMNPS)
608         WRITE(NDB) (ICONK(J, 1), J=1, NUMNPS)
609         WRITE(NDB) (ICONK(J, 1), J=1, LNPSNL)
610         WRITE(NDB) (OUT(J, 1), J=1, LNPSNL)
611         WRITE(NDB) (ICONK(J, 1), J=1, NUMESS)
612         WRITE(NDB) (ICONK(J, 1), J=1, NUMESS)
613         WRITE(NDB) (ICONK(J, 1), J=1, NUMESS)
614         WRITE(NDB) (ICONK(J, 1), J=1, NUMESS)
615         WRITE(NDB) (ICONK(J, 1), J=1, NUMESS)
616         WRITE(NDB) (ICONK(J, 1), J=1, LESSEL)
617         WRITE(NDB) (ICONK(J, 1), J=1, LESSNL)
618         WRITE(NDB) (OUT(J, 1), J=1, LESSNL)
619         WRITE (NDB) 1

```

```

620      WRITE (NDB) CNAME1,CNAME2,CDATE,CTIME
621      WRITE (NDB) 0
622      NAMECO(1)='X'
623      NAMECO(2)='Y'
624      WRITE (NDB) (NAMECO(I), I=1,NDIM)
625      NAMELB(1)='QUAD'
626      WRITE (NDB) NAMELB(1)
627 C
628      NAMEGV(1)='MACHINF'
629      NAMEGV(2)='BETA'
630      NVARGL=2
631
632      NAMENV(1)='DISPX'
633      NAMENV(2)='DISPY'
634      NAMENV(3)='ERRQT'
635      NAMENV(4)='U'
636      NAMENV(5)='V'
637      NAMENV(6)='Q'
638      NAMENV(7)='DENSITY'
639      NAMENV(8)='PRESSURE'
640      NAMENV(9)='ENERGY'
641      NAMENV(10)='MACH'
642      NAMENV(11)='DIVUV'
643      NAMENV(12)='DUDX'
644      NAMENV(13)='DVUV'
645      NAMENV(14)='DUDYVDX'
646      NAMENV(15)='MGGDUV'
647      NAMENV(16)='MGDVGDUV'
648      NAMENV(17)='NSUM1'
649      NAMENV(18)='NSUM2'
650      NAMENV(19)='NSUM3'
651      NAMENV(20)='NSUM4'
652      NAMENV(21)='KSUM1'
653      NAMENV(22)='KSUM2'
654      NAMENV(23)='KSUM3'
655      NAMENV(24)='KSUM4'
656      NVARNP=24
657      WRITE (NDB) 1,NVARGL,NVARNP,1
658      WRITE (NDB) 'EMPTYSET',(NAMEGV(I),I=1,NVARGL),
659 1      (NAMENV(I), I=1,NVARNP),'EMPTYSET'
660      WRITE (NDB) 0
661
662 C Output axis variable for use with GRAPH
663      runid(idlen:idlen+4)='.axs'
664      OPEN (UNIT=NAB,FILE=runid,STATUS='unknown',FORM='FORMATTED')
665      WRITE (NAB,('( ' !',A80)') CHAR
666      ENDIF
667
668      TIMEN=IBVARY+(IMVARY-1)*NBETA
669      WRITE (NDB) TIMEN, 0.
670      WRITE (NDB) (OUT(J,1),J=1,1)
671      OUT(1,1)=XMACHI
672      OUT(2,1)=BETAD
673      WRITE (NDB) (OUT(J,1),J=1,NVARGL)
674 C      must have NVARNP calls to WRTNV
675      CALL WRTNV(X,X0,2,XUNITS,NQPNTS,NTMAX,NTPNTS,IBETA,NDB,OUT)
676      CALL WRTNV(Y,Y0,2,XUNITS,NQPNTS,NTMAX,NTPNTS,IBETA,NDB,OUT)

```

```

677 CALL WRTNV(DIS, TMP, 1, XUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
678 CALL WRTNV(U, TMP, 1, VUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
679 CALL WRTNV(V, TMP, 1, VUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
680 CALL WRTNV(Q, TMP, 3, VUNITS, NQPNTS, 1, NTPNTS, IBETA, NDB, OUT)
681 CALL WRTNV(RHO, TMP, 3, DUNITS, NQPNTS, 1, NTPNTS, IBETA, NDB, OUT)
682 CALL WRTNV(PRES, TMP, 3, PUNITS, NQPNTS, 1, NTPNTS, IBETA, NDB, OUT)
683 CALL WRTNV(ENERGY, TMP, 3, EUNITS, NQPNTS, 1, NTPNTS, IBETA, NDB, OUT)
684 CALL WRTNV(XMACH2, TMP, 4, 1., NQPNTS, 1, NTPNTS, IBETA, NDB, OUT)
685 CALL WRTNV(DIVU, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
686 CALL WRTNV(UX, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
687 CALL WRTNV(VY, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
688 CALL WRTNV(UYORVX, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
689 CALL WRTNV(XMAGDU, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
690 CALL WRTNV(DEVU, TMP, 1, GUNITS, NQPNTS, NTMAX, NTPNTS, IBETA, NDB, OUT)
691 DO 4950 NK=1,2
692 DO 4950 L=1,4
693 ITMP=0
694 DO 4900 J=1, NQPNTS
695 DO 4900 I=1, NTPNTS
696 IF (J.EQ. NQPNTS.AND.
697 1 (I.EQ.1.OR.I.EQ.NTPNTS.OR.I.EQ.IBETA))GOTO4900
698 IF (NK.EQ.1) TMP=NEST(I,J,INTFLG,L)
699 IF (NK.EQ.2) TMP=KEST(I,J,INTFLG,L)
700 ITMP=ITMP+1
701 OUT(ITMP,1)=TMP
702 4900 CONTINUE
703 WRITE(NDB) (OUT(I,1),I=1,ITMP)
704 4950 CONTINUE
705
706 C On axis output for line graph
707
708 WRITE(NAB,4975) TIMEN,XMACHI,BETAD
709 4975 FORMAT(' !OUTPUT NUMBER = ',F5.0,
710 1 ' ', MACH = ',F10.4,', BETA = ',F10.2,/,
711 1 ' ! X $ U $ RHO $ ',
712 2 ' PRES $ ENERGY $ MACHNO $ ',
713 3 ' DIVU $ UX $ VY $ ',
714 4 ' XMAGDU $ DEVU $')
715 DO 4980 I=NTPNTS,1,-NTPNTS+1
716 IF (I.EQ. NTPNTS) THEN
717 JMIN=NQPNTS-1
718 JMAX=1
719 JIT=-1
720 ELSE
721 JMIN=1
722 JMAX=NQPNTS-1
723 JIT=1
724 ENDIF
725 DO 4980 J=JMIN,JMAX,JIT
726 OUT(1,1)=X(I,J)*XUNITS
727 OUT(2,1)=U(I,J)*VUNITS
728 OUT(3,1)=RHO(J)*DUNITS
729 OUT(4,1)=PRES(J)*PUNITS
730 OUT(5,1)=ENERGY(J)*PUNITS
731 OUT(6,1)=0.
732 IF (J.NE.1) OUT(6,1)=SQRT(XMACH2(J))
733 OUT(7,1)=DIVU(I,J)*GUNITS

```



```

791 C      NMAX ; maximum number of terms from series
792 C      ATYPE ; Character variable giving the requested
793 C           acceleration scheme.
794 C           Options : NONE
795 C           EPSILON (KMAX must be odd. Also known as SHANKS
796 C                   transformation for KMAX = 3 and iterated
797 C                   SHANKS for KMAX greater than 3. Roughly
798 C                   half of the significant figures will be
799 C                   lost using this algorithm.)
800 C           THETA (KMAX must be odd.)
801 C           LEVINU KMAX must be GE 2
802 C           LEVINT KMAX must be GE 2
803 C
804 C      Output:
805 C      SEST ; best estimate for sum
806 C      NEST ; row number of Lozenge giving best estimate
807 C      KEST ; column number of Lozenge giving estimate (usually KMAX)
808 C      S(NMAX,KMAX) ; Complete Lozenge diagram. S must be dimensioned
809 C                   as S(NDIM,K) where K .GE. KMAX.
810 C      IERR ; 0 for successful completion; 1 for unsuccessful
811 C
812 C
813      SUBROUTINE CSUM(A,S,NDIM,SEST,RELERR,NEST,KEST,NMAX,KMAX,
814      1              ATYPE,IERR)
815      IMPLICIT COMPLEX (A-H,O-Z)
816      REAL RELERR
817      DIMENSION S(NDIM,KMAX)
818      PARAMETER (MAXQ=1000)
819      DIMENSION QN(MAXQ),QD(MAXQ)
820      CHARACTER*(*) ATYPE
821      EXTERNAL A
822      IERR=0
823      IF(ATYPE.EQ.'NONE') THEN
824 C      ignore KMAX
825      KEST = 1
826      S(1,1) = A(1)
827      S(2,1) = S(1,1)+A(2)
828      DO 20 N=3,NMAX
829      S(N,1)= S(N-1,1)+A(N)
830      IF(S(N,1).EQ.0. .AND. S(N-1,1).EQ.0. .AND.
831      1      S(N-2,1).EQ.0. ) GOTO 30
832      IF(ABS( S(N,1)-S(N-1,1))/S(N,1)) .LE. RELERR. AND.
833      1      ABS( S(N,1)-S(N-2,1))/S(N,1)) .LE. RELERR) GOTO 30
834 20  CONTINUE
835      WRITE(6,*) 'NO CONVERGENCE IN SUM. BEST ESTIMATE WILL BE USED.'
836      IERR=1
837      N=NMAX
838 30  SEST = S(N,1)
839      NEST = N
840      ELSEIF(ATYPE.EQ.'EPSILON') THEN
841 C      KMAX = 1 implies simple summation.
842      IF(MOD(KMAX,2).NE.1.) THEN
843      WRITE(6,*) 'KMAX must be odd for epsilon algorithm.'
844      STOP
845      ENDIF
846      S(1,1)=A(1)
847      DO 200 N=2,NMAX

```

```

848      S(N,1)=A(N)+S(N-1,1)
849      KEST=MIN(N,KMAX)
850      DO 100 K=2,KEST
851          TMP1 = S(N-K+2,K-1)-S(N-K+1,K-1)
852 C      Assume no zero terms so that consecutive equality
853 C      implies convergence.
854          IF(TMP1.EQ.0.) THEN
855              NEST=N-K+2
856              KEST=K-1
857              SEST=S(NEST,KEST)
858              GOTO220
859          ENDIF
860          TMP2=0.
861          IF(K.GE.3) TMP2=S(N-K+2,K-2)
862          S(N-K+1,K) = TMP2 + 1./TMP1
863 100    CONTINUE
864          IF(MOD(N,2).EQ.1) THEN
865              NEST = N-KEST+1
866              SEST = S(NEST,KEST)
867              SEST1 = S(MIN(NEST+2,N),MAX(KEST-2,1))
868              SEST2 = S(MIN(NEST+1,N-1),MAX(KEST-2,1))
869              IF(ABS((SEST-SEST1)/SEST).LE.RELERR.AND.
870 1        ABS((SEST-SEST2)/SEST).LE.RELERR) GOTO 220
871          ENDIF
872 200    CONTINUE
873          WRITE(6,*) 'NO CONVERGENCE IN EPSILON ALGORITHM SUM.'
874          WRITE(6,*) 'BEST ESTIMATE WILL BE USED.'
875          IERR=1
876 220    CONTINUE
877      ELSEIF(ATYPE.EQ.'THETA') THEN
878 C      KMAX = 1 implies simple summation.
879          IF(MOD(KMAX,2).NE.1.) THEN
880              WRITE(6,*) 'KMAX must be odd for THETA algorithm.'
881              STOP
882          ENDIF
883          S(1,1)=A(1)
884          S(2,1)=S(1,1)+A(2)
885          DO 1200 N=3,NMAX-1,2
886              KTMP=MIN(N/2+1,KMAX)
887          DO 1200 J=0,1
888              S(N+J,1)=S(N-1+J,1)+A(N+J)
889              NEST=N+J
890              KEST=1
891
892              NUP=N+J
893              K=1
894              DO 1100 K=2,KTMP
895                  NUP=N-2*K+2+J
896                  IF(MOD(K,2).EQ.0) THEN
897 C      update even column: not an estimate
898                      TMP1 = S(NUP+1,K-1)-S(NUP,K-1)
899                      IF(TMP1.EQ.0.) THEN
900                          NEST = NUP+1
901                          KEST = K-1
902                          SEST = S(NEST,KEST)
903                          GOTO 1220
904                      ENDIF

```

```

905             TMP2=0.
906             IF(K.GE.4) TMP2=S(NUP+1,K-2)
907             S(NUP,K) = TMP2 + 1./TMP1
908             ELSE
909 C             update odd columns giving estimates
910             TMP1=S(NUP+1,K-1)-S(NUP,K-1)
911             TMP2=S(NUP+2,K-1)-S(NUP+1,K-1)
912             TMP3=TMP2-TMP1
913             S(NUP,K)=(S(NUP+2,K-2)*TMP2-S(NUP+1,K-2)*TMP1)/TMP3
914             KEST=K
915             NEST=NUP
916             ENDIF
917 1100        CONTINUE
918             SEST=S(NEST,KEST)
919             IF(KEST.EQ.1) THEN
920                 SEST1=S(NEST-1,KEST)
921                 SEST2=S(NEST-2,KEST)
922             ELSEIF(NEST.EQ.1) THEN
923                 SEST1=S(NEST+3,KEST-2)
924                 SEST2=S(NEST+4,KEST-2)
925             ELSE
926                 SEST1=S(NEST-1,KEST)
927                 SEST2=S(NEST+4,KEST-2)
928             ENDIF
929             IF(ABS((SEST-SEST1)/SEST).LE.RELERR.AND.
930 1           ABS((SEST-SEST2)/SEST).LE.RELERR) GOTO 1220
931 1200        CONTINUE
932             WRITE(6,*) 'NO CONVERGENCE IN THETA ALGORITHM SUM.'
933             WRITE(6,*) 'BEST ESTIMATE WILL BE USED.'
934             IERR=1
935 1220        CONTINUE
936             ELSEIF(ATYPE.EQ.'LEVINU'.OR.ATYPE.EQ.'LEVINT') THEN
937                 IF(KMAX.GT.MAXQ.OR.KMAX.LT.2) STOP
938                 S(1,1) = A(1)
939                 AN=A(2)
940 C Zero terms imply convergence
941                 IF(AN.EQ.0.) THEN
942                     NEST=1
943                     KEST=1
944                     SEST=S(NEST,KEST)
945                     GOTO 3000
946                 ENDIF
947                 QD(1) = 1./AN
948                 QN(1)=S(1,1)*QD(1)
949                 DO 2000 N=2,NMAX-1
950                     S(N,1)= S(N-1,1)+AN
951                     AN=A(N+1)
952                     IF(AN.EQ.0.) THEN
953                         NEST=N
954                         KEST=1
955                         SEST=S(NEST,KEST)
956                         GOTO 3000
957                     ENDIF
958                     QD(N) = 1./AN/N
959                     IF(ATYPE.EQ.'LEVINU') QD(N)=QD(N)/N
960                     QN(N)=S(N,1)*QD(N)
961                 DO 2000 K = 2,MIN(N,KMAX)

```

```

962         NMKP1=N-K+1
963         TMP1 = 1.
964         IF(K.GT.2.) TMP1= (FLOAT(N-1)/FLOAT(N))**(K-2)
965         TMP1 = FLOAT(NMKP1)/N*TMP1
966         QN(NMKP1) = QN(NMKP1+1) - TMP1*QN(NMKP1)
967         QD(NMKP1) = QD(NMKP1+1) - TMP1*QD(NMKP1)
968         TMP=QD(NMKP1)
969         IF(TMP.EQ.0.) TMP=1.E-15
970         S(NMKP1,K) = QN(NMKP1)/TMP
971         NEST = NMKP1
972         KEST = K
973         SEST=S(NEST,KEST)
974         SEST1=S(NEST+1,KEST-1)
975         SEST2=S(NEST,KEST-1)
976         IF(ABS((SEST-SEST1)/SEST).LE.RELERR.AND.
977         1     ABS((SEST-SEST2)/SEST).LE.RELERR) GOTO 3000
978 2000     CONTINUE
979         WRITE(6,*) 'NO CONVERGENCE IN ',ATYPE,' SUM.'
980         WRITE(6,*) 'BEST ESTIMATE WILL BE USED.'
981         IERR=1
982 3000     CONTINUE
983     ELSE
984         WRITE(6,*) 'BAD VALUE OF ACCELERATION FLAG IN ROUTINE SUM'
985         STOP
986     ENDIF
987     RETURN
988     END
989
990     COMPLEX FUNCTION ZN(I)
991     PARAMETER(NQMAX=100,NUMAX=200,NTMAX=100)
992     COMPLEX EITHTA
993     COMMON /FLOCOM/ IQ,IQ1,EITHTA,PI,SNTH(NUMAX),CNTH(NUMAX)
994     COMMON /GASCOM/ GAMMA,GM1D2,QMAX2,QCR2,TAUCR,Q1,TAU1,PSI2T1,
995     1     Q(NQMAX),TAU(NQMAX),RHO(NQMAX),
996     2     ENERGY(NQMAX),PRES(NQMAX),XMACH2(NQMAX),
997     3     FPDF(NUMAX,NQMAX),XLNF(NUMAX,NQMAX),
998     4     F(NUMAX,NQMAX),FP(NUMAX,NQMAX),
999     5     PSI(NUMAX,NQMAX),PSIP(NUMAX,NQMAX),
1000    6     PSIRAT(NUMAX,NQMAX),NCHPSM(NUMAX,NQMAX)
1001     IF(I.EQ.1) THEN
1002         ZN=0.
1003     ELSE
1004         IF(IQ.EQ.1) THEN
1005 C         Q=0 value
1006         ZN =0.
1007     ELSE
1008 C         Q ne zero
1009         TMP3=-(1./I + PSIRAT(I,IQ))
1010         TMP4=1.+PSIRAT(I,IQ)/I
1011         TMP5=I/FLOAT(I*I-1)
1012         TMP5=TMP5*PSI(I,IQ)/(RHO(IQ)*Q(IQ))
1013         ZN =EITHTA*CMPLX(TMP3*CNTH(I),TMP4*SNTH(I))*TMP5
1014     ENDIF
1015     ENDIF
1016     ZN=ZN*Q1*RHO(IQ1)/PI
1017     RETURN
1018     END

```

```

1019
1020     COMPLEX FUNCTION ZDQ(I)
1021     PARAMETER(NQMAX=100,NUMAX=200,NTMAX=100)
1022     COMMON /FLOCOM/ IQ,IQ1,EITHTA,PI,SNTH(NUMAX),CNTH(NUMAX)
1023     COMMON /GASCOM/ GAMMA,GM1D2,QMAX2,QCR2,TAUCR,Q1,TAU1,PSI2T1,
1024     1           Q(NQMAX),TAU(NQMAX),RHO(NQMAX),
1025     2           ENERGY(NQMAX),PRES(NQMAX),XMACH2(NQMAX),
1026     3           FPDF(NUMAX,NQMAX),XLNF(NUMAX,NQMAX),
1027     4           F(NUMAX,NQMAX),FP(NUMAX,NQMAX),
1028     5           PSI(NUMAX,NQMAX),PSIP(NUMAX,NQMAX),
1029     6           PSIRAT(NUMAX,NQMAX),NCHPSM(NUMAX,NQMAX)
1030     COMPLEX EITHTA
1031     IF(I.EQ.1) THEN
1032         ZDQ=0.
1033     ELSE
1034         IF(IQ.EQ.1) THEN
1035 C           Q=0 LIMITS
1036             IF(I.EQ.2) THEN
1037                 TMP2=1./(QMAX2*PSI2T1)
1038             ELSE
1039                 TMP2=0.
1040             ENDIF
1041         ELSE
1042 C           Q NE 0
1043             TMP2=PSI(I,IQ)/(RHO(IQ)*Q(IQ)**2)
1044         ENDIF
1045         TMP3=-(.1.-XMACH2(IQ))
1046         TMP4=PSIRAT(I,IQ)
1047         ZDQ =TMP2*EITHTA*CMPLX(TMP3*CNTH(I),TMP4*SNTH(I))
1048         ZDQ=ZDQ*Q1*RHO(IQ1)/PI
1049     ENDIF
1050     RETURN
1051     END
1052
1053     SUBROUTINE CHPLGN(NQPNTS,RELERR)
1054 C     Compute important Chaplygin function quantities:
1055 C         F'/F (FPDF), log F (XLNF), F (F), F' (FP),
1056 C         \psi/\psi(\tau_1) (PSI), \psiip/\psi(\tau_1) (PSIP)
1057 C         2\tau \psiip/(n\psi) (PSIRAT)
1058 C         \psi_2(\tau_1) (PSI2T1)
1059 C     for NQPNTS \tau points and positive integral order up to NUMAX to
1060 C     a relative error of RELERR. TAU(1) must be 0 and TAU(NQPNTS)=TAU1.
1061     PARAMETER(NQMAX=100,NUMAX=200,NTMAX=100)
1062     PARAMETER(NCMAX=500)
1063     DIMENSION AS(NCMAX),BS(NCMAX),SUM(NCMAX),Z(11)
1064 C     Continued Fraction Sum COMmon passes parameters to CFSUM
1065     COMMON/CFSCOM/ CFSPRM(4)
1066     COMMON /GASCOM/ GAMMA,GM1D2,QMAX2,QCR2,TAUCR,Q1,TAU1,PSI2T1,
1067     1           Q(NQMAX),TAU(NQMAX),RHO(NQMAX),
1068     2           ENERGY(NQMAX),PRES(NQMAX),XMACH2(NQMAX),
1069     3           FPDF(NUMAX,NQMAX),XLNF(NUMAX,NQMAX),
1070     4           F(NUMAX,NQMAX),FP(NUMAX,NQMAX),
1071     5           PSI(NUMAX,NQMAX),PSIP(NUMAX,NQMAX),
1072     6           PSIRAT(NUMAX,NQMAX),NCHPSM(NUMAX,NQMAX)
1073     DO 400 NU=1,NUMAX
1074         APB=NU-1./(GAMMA-1.)
1075         ATB=-.5*NU*(NU+1)/(GAMMA-1.)

```

```

1076      B=(APB+SQRT(APB**2-4.*ATB))/2.
1077      A=ATB/B
1078      C=NU+1
1079      CFSPRM(1)=A
1080      CFSPRM(2)=B
1081      CFSPRM(3)=C
1082      CFSPRM(4)=0
1083      IF(TAU(1).NE.0.) THEN
1084          WRITE(6,*) 'TAU(1) .NE. 0'
1085          STOP
1086      ENDIF
1087      CALL CFSUM(AS,BS,SUM,NSUM,NCMAX,RELERR)
1088      NCHPSM(NU,1)=NSUM
1089      TMP = C*SUM(NSUM)/ATB
1090      TMP2= 1./TMP
1091      FPDF(NU,1)=TMP2
1092      XLNF(NU,1)=0.
1093      DO 400 I = 2,NQPNTS
1094          TMP1=TMP2
1095          XL = TAU(I-1)
1096          XR = TAU(I)
1097          H = XR-XL
1098          CFSPRM(4)=XR
1099          CALL CFSUM(AS,BS,SUM,NSUM,NCMAX,RELERR)
1100          NCHPSM(NU,I)=NSUM
1101          TMP = C*SUM(NSUM)/A
1102          ZTAU = TAU(I)
1103      C **** ZTAU is zero for second continued fraction
1104          IF(ZTAU.LT.0.) ZTAU=0.
1105          TMP = (TMP-ZTAU)/B
1106          TMP2= 1./TMP
1107          FPDF(NU,I)=TMP2
1108          TMP3 =.5*(TMP1+TMP2)
1109          Z(1) = TMP3*H
1110
1111          DO 395 K=1,10
1112              KNUM=2**K
1113              DO 390 L=1,KNUM/2
1114                  XTMP = H*FLOAT(2*L-1)/KNUM + XL
1115                  CFSPRM(4)=XTMP
1116                  CALL CFSUM(AS,BS,SUM,NSUM,NCMAX,RELERR)
1117                  TMP = C*SUM(NSUM)/A
1118                  ZTAU = XTMP
1119      C **** ZTAU is zero for second continued fraction
1120          IF(ZTAU.LT.0.) ZTAU=0.
1121          TMP = (TMP-ZTAU)/B
1122          TMP = 1./TMP
1123          TMP3 = TMP3+TMP
1124      390      CONTINUE
1125          Z(K+1)=TMP3*H/KNUM
1126          TMP=KNUM**2
1127          DO 391 L=K,1,-1
1128      391      Z(L)= (TMP*Z(L+1)-Z(L))/(TMP-1.)
1129          IF(ABS( (Z(1)-Z(2))/Z(1) ).LT.RELERR) GOT0399
1130      395      CONTINUE
1131          WRITE(6,*) 'NO ROMBERG CONVERGENCE'
1132          STOP

```

```

1133 399      XLNF(NU,I) = XLNF(NU,I-1) + Z(1)
1134 400      CONTINUE
1135
1136 C      Evaluate remaining quantities
1137 C      \tau = 0
1138 C      Order 1
1139          NU=1
1140          F(NU,1)=1.
1141          FP(NU,1)=FPDF(NU,1)
1142          PSI(NU,1)=0.
1143 C      d\psi_1/d\tau is infinite at \tau = 0.
1144          PSIP(NU,1)=1.E+30
1145          PSIRAT(NU,1)=1.
1146 C      Order 2
1147          NU=2
1148          F(NU,1)=1.
1149          FP(NU,1)=FPDF(NU,1)
1150          PSI(NU,1)=0.
1151          PSI2T1=TAU(NQPNTS)*EXP(XLNF(NU,NQPNTS))
1152          PSIP(NU,1)=1./PSI2T1
1153          PSIRAT(NU,1)=1.
1154 C      Order 3 or move
1155          DO 500 NU=3,NUMAX
1156              F(NU,1)=1.
1157              FP(NU,1)=FPDF(NU,1)
1158              PSI(NU,1)=0.
1159              PSIP(NU,1)=0.
1160              PSIRAT(NU,1)=1.
1161 500      CONTINUE
1162 C      \tau greater than zero
1163          DO 600 I=2,NQPNTS
1164              DO 600 NU=1,NUMAX
1165                  F(NU,I)=EXP(XLNF(NU,I))
1166                  FP(NU,I)=FPDF(NU,I)*F(NU,I)
1167 C      PSI's must be normalized to avoid underflow at large nu
1168 C      Thus PSI = \psi(\tau)/\psi(\tau_1) and PSIP = \psi'(\tau)/\psi(\tau_1)
1169          PSI(NU,I)=EXP(.5*NU*LOG(TAU(I)/TAU(NQPNTS))
1170              1          +XLNF(NU,I)-XLNF(NU,NQPNTS))
1171          TMP=2.*TAU(I)/NU
1172          PSIP(NU,I)=PSI(NU,I)*(1./TMP+FPDF(NU,I))
1173          PSIRAT(NU,I)=1.+TMP*FPDF(NU,I)
1174 600      CONTINUE
1175
1176          RETURN
1177          END
1178
1179 C      Routine to evaluate a continued fraction of the form
1180 C      b_0 + a_1/b_1 + a_2/b_2 + ...
1181 C      with numerator A_n and denominator B_n of the n_th approximate
1182 C      where
1183 C      A_n = b_n*A_{n-1} + a_n * A_{n-2}
1184 C      B_n = b_n*B_{n-1} + a_n * B_{n-2}
1185 C
1186 C      for n = 1,2,...
1187 C
1188 C      A_{-1}=1 ,  A_0 = b_0 ,  B_{-1}=0, B_0=1
1189 C

```

```

1190 C      Input:
1191 C          NMAX = Maximum value of N. N must be at least 3
1192 C          RELERR = Desired relative error between successive approximations
1193 C                   before stopping. When 3 successive estimates vary by less
1194 C                   than RELERR from their mean, then the evaluation process stops.
1195 C      Output:
1196 C          NSUM = value of n at which evaluation stops
1197 C          (A(I),I=1,NSUM) = values of the coefficients A_n
1198 C          (B(I),I=1,NSUM) = values of the coefficients B_n
1199 C          (SUM(I),I=1,NSUM) = successive estimates for value of continued
1200 C                   fraction.
1201 C
1202 C      W is a convergence acceleration factor for limit periodic continued
1203 C      fractions. W is set equal to the subdominant root of
1204 C      S**2-b*S-a=0 in ABFUNC to subtract out leading subdominant term.
1205 C      Otherwise W=0.
1206 C
1207 C      Warning: This routine does not test for divide by zero.
1208 C
1209 C      SUBROUTINE CFSUM(A,B,SUM,NSUM,NMAX,RELERR)
1210 C      DIMENSION A(NMAX),B(NMAX),SUM(NMAX)
1211 C      PARAMETER(RESCALE=1.E+25)
1212 C      CALL ABFUNC(0,ADUM,BO,W)
1213 C      CALL ABFUNC(1,A1,B1,W)
1214 C      A(1)=B1*BO + A1
1215 C      B(1)=B1
1216 C      SUM(1)=A(1)/B(1)
1217 C      CALL ABFUNC(2,A2,B2,W)
1218 C      A(2)=B2*A(1)+A2*BO
1219 C      B(2)=B2*B(1)+A2
1220 C      SUM(2)=(A(2)-W*A(1))/(B(2)-W*B(2))
1221 C      DO 100 I=3,NMAX
1222 C          CALL ABFUNC(I,AI,BI,W)
1223 C          IF (ABS(A(I-1)).GT.RESCALE .OR.
1224 C      1      ABS(B(I-1)).GT.RESCALE ) THEN
1225 C              A(I-1)=A(I-1)/RESCALE
1226 C              A(I-2)=A(I-2)/RESCALE
1227 C              B(I-1)=B(I-1)/RESCALE
1228 C              B(I-2)=B(I-2)/RESCALE
1229 C          ENDIF
1230 C          A(I)=BI*A(I-1)+AI*A(I-2)
1231 C          B(I)=BI*B(I-1)+AI*B(I-2)
1232 C          SUM(I)=(A(I)-A(I-1)*W)/(B(I)-B(I-1)*W)
1233 C          SEST=(SUM(I)+SUM(I-1)+SUM(I-2))/3.
1234 C          TMP=ABS(SUM(I)/SEST-1.)
1235 C          IF( TMP .LE.RELERR .AND.
1236 C      1      ABS(SUM(I-1)/SEST-1.) .LE.RELERR .AND.
1237 C      2      ABS(SUM(I-2)/SEST-1.) .LE.RELERR) THEN
1238 C              NSUM=I
1239 C              RETURN
1240 C          ENDIF
1241 C      100 CONTINUE
1242 C      WRITE(6,*) 'CONVERGENCE NOT ACHIEVED IN CFSUM'
1243 C      NSUM=NMAX
1244 C      RETURN
1245 C      END
1246 C

```

```

1247 C Evaluate A_n and B_n for hypergeometric function continued fractions.
1248 C ABFUNC is called by CFSUM.
1249 C W is the subdominant root of the limit characteristic equation and
1250 C can significantly accelerate the convergence for some continued fractions.
1251 C In this application the acceleration option is not implemented, i.e. W=0.
1252     SUBROUTINE ABFUNC(N,AN,BN,W)
1253     COMMON/CFSCOM/ CFSPRM(4)
1254     A=CFSPRM(1)
1255     B=CFSPRM(2)
1256     C=CFSPRM(3)
1257     X=CFSPRM(4)
1258     IF (X.GE.0.) THEN
1259         AN=-(B+N)*(C-A+N-1)/((C+N-1)*(C+N))
1260         AN=AN*X
1261         BN=(B-A+N)/(C+N)
1262         BN=BN*X+1
1263         IF (N.EQ.0) BN=1.
1264         W=0.
1265 C     W=X ! Subdominant root of limit char. equation for acceleration.
1266 C     In this case the asymptotic behavior of AN and BN only goes like
1267 C     1/N so the convergence acceleration is not very spectacular
1268 C     especially for large N. Convergence acceleration is not implemented.
1269     ELSE
1270         AN=(B+N)*(A+N)/((C+N-1)*(C+N))
1271         AN=AN*X*(1.-X)
1272         BN=(A+B+2*N+1)/(C+N)
1273         BN=1.-BN*X
1274         W=0.
1275 C     W=-X ! Subdominant root of limit char. equation for acceleration.
1276 C     In this case the asymptotic behavior of AN and BN only goes like
1277 C     1/N so the convergence acceleration is not very spectacular
1278 C     especially for large N. Convergence acceleration is not implemented.
1279     ENDIF
1280     RETURN
1281     END
1282
1283 C Modified Hermite interpolation fits an NF+NDF-1 degree polynomial to
1284 C NF points of a function and the derivative of the function at NDF
1285 C of these points. Hermite interpolation implies NF=NDF.
1286 C This routine evaluates the modified Hermite interpolation polynomial
1287 C at the point X.
1288 C Reference: A. Ralston and P. Rabinowitz, A First Course in Numerical
1289 C Analysis, McGraw-Hill, 1978, pp.70-73.
1290 C Author: Allen C. Robinson
1291 C Last modification date: October 14, 1988
1292     SUBROUTINE HERMIT(XI,FI,DFI,NF,NDF,X,F,DF)
1293 C INPUT
1294 C     NF = NUMBER OF INTERPOLATION POINTS (NF .LE. 20)
1295 C     NDF = NUMBER OF DERIVATIVES TO INTERPOLATE
1296 C     XI(I) = ARRAY OF DISTINCT ABSCISSA POINTS I=1,NF
1297 C     FI(I) = FUNCTION VALUES AT XI(I),I=1,NF
1298 C     DFI(I) = DERIVATIVE VALUES AT XI(I),I=1,NDF WHERE NDF<=NF
1299 C     X = ABSCISSA FOR INTERPOLATION
1300 C OUTPUT
1301 C     F = VALUE OF INTERPOLANT AT X
1302 C     DF = VALUE OF DERIVATIVE OF INTERPOLANT AT X
1303 C     ERROR MESSAGES TO UNIT 6

```

```

1304 C
1305     REAL WORK(20),XI(NF),FI(NF),DFI(NDF)
1306     IF(NF.GT.20) THEN
1307         WRITE(6,*) 'TOO MANY INTERPOLATION POINTS'
1308         STOP
1309     ENDIF
1310 C Check for X equal one of interpolation points with given derivative.
1311     DO 5 I=1,NDF
1312         IF(X.EQ.XI(I)) THEN
1313             F=FI(I)
1314             DF=DFI(I)
1315             GOTO150
1316         ENDIF
1317 5     CONTINUE
1318 C Check for X equal to a simple interpolation point.
1319     DO 10 II=NDF+1,NF
1320         IF(X.EQ.XI(II)) THEN
1321             F=FI(II)
1322             DF=0
1323             DO 15 J=1,NF
1324                 PJNP=1.
1325                 PJNPJ=0.
1326                 DO 16 I=1,NF
1327                     IF(I.NE.J) THEN
1328                         TMP=1./(XI(J)-XI(I))
1329                         IF(I.NE.II) PJNP=PJNP*(X-XI(I))*TMP
1330                         PJNPJ=PJNPJ+TMP
1331                     ENDIF
1332                     IF(I.EQ.NDF) THEN
1333                         PJR=PJNP
1334                         PJRPJ=PJNPJ
1335                     ENDIF
1336 16                 CONTINUE
1337                 TMP4=0.
1338                 IF(J.EQ.II) THEN
1339                     PJNP=PJNPJ
1340                     DO 17 K=1,NDF
1341 17                     TMP4=TMP4+PJR/(X-XI(K))
1342                 ELSE
1343                     PJNP=PJNP/(XI(J)-XI(II))
1344                 ENDIF
1345                 IF(J.LE.NDF) THEN
1346                     TMP=X-XI(J)
1347                     TMP3=PJNP*PJR
1348                     DF=DF+(1.-TMP*(PJNPJ+PJRPJ))*TMP3*FI(J)
1349                     DF=DF+TMP*TMP3*DFI(J)
1350                 ELSE
1351                     DF=DF+(PJNP*PJR+TMP4)*FI(J)
1352                 ENDIF
1353 15                 CONTINUE
1354                 GOTO150
1355             ENDIF
1356 10     CONTINUE
1357 C Evaluate polynomial and derivative and X
1358     F=0
1359     DF=0
1360     DO 50 J=1,NDF

```

```

1361         PJN=1.
1362         PJNPJ=0.
1363         DO 25 I=1,NF
1364             IF(I.NE.J) THEN
1365                 TMP=1./(XI(J)-XI(I))
1366                 WORK(I)=X-XI(I)
1367                 PJN=PJN*WORK(I)*TMP
1368                 PJNPJ=PJNPJ+TMP
1369             ENDIF
1370             IF(I.EQ.NDF) THEN
1371                 PJR=PJN
1372                 PJRPJ=PJNPJ
1373             ENDIF
1374 25        CONTINUE
1375         PJNP=0.
1376         PJRP=0.
1377         DO 30 I=1,NF
1378             IF(I.NE.J) PJNP=PJNP+PJN/WORK(I)
1379             IF(I.NE.J.AND.I.LE.NDF) PJRP=PJRP+PJR/WORK(I)
1380 30        CONTINUE
1381         TMP=X-XI(J)
1382         TMP1=PJN*PJR
1383         TMP2=PJNPJ+PJRPJ
1384         TMP3=1.-TMP*TMP2
1385         F=F+TMP3*TMP1*FI(J)
1386         F=F+TMP*TMP1*DFI(J)
1387         TMP4=PJNP*PJR+PJN*PJRP
1388         DF=DF+FI(J)*(-1.*TMP2*TMP1+TMP3*TMP4)
1389         DF=DF+DFI(J)*(TMP1+TMP*TMP4)
1390 50        CONTINUE
1391         DO 100 J=NDF+1,NF
1392             PJN=1.
1393             DO 75 I=1,NF
1394                 IF(I.NE.J) THEN
1395                     TMP=1./(XI(J)-XI(I))
1396                     WORK(I)=X-XI(I)
1397                     PJN=PJN*WORK(I)*TMP
1398                 ENDIF
1399                 IF(I.EQ.NDF) PJR=PJN
1400 75        CONTINUE
1401             PJNP=0.
1402             PJRP=0.
1403             DO 80 I=1,NF
1404                 IF(I.NE.J) PJNP=PJNP+PJN/WORK(I)
1405                 IF(I.LE.NDF) PJRP=PJRP+PJR/WORK(I)
1406 80        CONTINUE
1407             TMP1=PJN*PJR
1408             F=F+PJN*PJR*FI(J)
1409             DF=DF+FI(J)*(PJNP*PJR+PJN*PJRP)
1410 100       CONTINUE
1411 150       RETURN
1412         END

```

Distribution

External Distribution:

Army Research Laboratory

Attn:

K. Kimsey

S. B. Segletes

W. P. Walters

Aberdeen Proving Ground

Aberdeen, MD 21005

L. Pamela Cook

Department of Mathematical Sciences

Univ. of Delaware

Newark, DE 19716

George S. Dulikravich

Department of Mechanical and Aerospace
Engineering

The Univ. of Texas at Arlington

Arlington, TX 76019

A. Friedman

Institute for Mathematics and its Applica-
tions

University of Minnesota

Minneapolis, MN 55455-0436

S. I. Hariharan

Dept. of Math. Sciences

Univ. of Akron

Akron, OH 44325-4002

James Glimm

OP-138A Math Tower

SUNY at Stony Brook

Stony Brook, NY 11794-3600

Lawrence Livermore National Laboratory

Attn:

T. F. Adams, L-095

Francois Heuze, L-200

Michael J. Murphy, L-099

Ray Pierce , MS L-160

Peter W. Rambo, MS L-015

University of California

Lawrence Livermore National Laboratory

7000 East Ave.

P.O. Box 808

Livermore, CA 94550

Los Alamos National Laboratory

Attn:

W. Rider, D413

J. Chapyak, F664

J. Hopson, F664

R. Karpp, P940

L. Schwalbe, F613

M. Shashkov, B284

P. O. Box 1663

Los Alamos, NM 87545

Srdjan Stojanovic

University of Cincinnati

Dept. Mathematical Science

Old Chemistry Bldg. (ML25)

Cincinnati, OH 45221-0025

Andrea Malagoli

The University of Chicago

Enrico Fermi Institute

933 E. 56th St.

LASR 132b

Chicago, Il 60637

Daniel I. Meiron , 08-31

Philip G. Saffman, 217-50

California Inst. of Technology

Pasadena, CA 91125

Bryan Oliver

Mission Research Corporation

1720 Randolph Rd.

Albuquerque, NM 87106

Internal Distribution:

MS 1186 T. Mehlhorn, 1674
MS 1186 T. Haill, 1674
MS 0836 M. R. Baer, 9100
MS 0826 D. K. Gartling, 9100
MS 0834 A. Ratzel, 9110
MS 0835 S. N. Kempka, 9113
MS 0825 W. L. Oberkampf, 9115
MS 0847 H. Morgan, 9120
MS 0835 J. S. Peery, 9121
MS 0865 J. Moya, 9130
MS 0828 M. Pilch, 9133
MS 0835 J. M. McGlaun, 9140
MS 0321 W. J. Camp, 9200
MS 0819 T. G. Trucano, 9211
MS 1110 P. Bochev, 9213
MS 1110 D. Womble, 9214
MS 1110 L. A. Romero, 9214
MS 1110 R. Lehoucq, 9222
MS 0820 P. Yarrington, 9230
MS 0819 E. Boucheron, 9231
MS 0819 K. H. Brown, 9231
MS 0819 K. G. Budge, 9231
MS 0819 D. Carroll, 9231
MS 0819 M. Christon, 9231
MS 0819 R. Drake, 9231
MS 0819 C. J. Garasi, 9231
MS 0819 S. V. Petney, 9231
MS 0819 J. Robbins, 9231
MS 0819 A. C. Robinson, 9231 (25)
MS 0819 R. Summers, 9231
MS 0819 M. Wong, 9231
MS 0820 P. F. Chavez, 9232
MS 0820 R. Weatherby, 9231
MS 0820 D. Crawford, 9232
MS 0820 A. Farnsworth, 9232
MS 0820 M. E. Kipp, 9232
MS 0820 S. A. Silling, 9232
MS 0820 P. A. Taylor, 9232
MS 0316 S. S. Dosanjh, 9233
MS 1111 A. Salinger, 9233
MS 1111 J. N. Shadid, 9233
MS 1111 R. Schmidt, 9233
MS 1111 T. M. Smith, 9233
MS 0316 J. B. Aidun, 9235
MS 9018 Central Technical Files, 8945-1 (1)
MS 0899 Technical Library, 9616 (2)
MS 0612 Review and Approval Desk, 9612
(1)
For DOE/OSTI